



# Variable surrogate model-based particle swarm optimization for high-dimensional expensive problems

Jie Tian<sup>1,2</sup> · Mingdong Hou<sup>3</sup> · Hongli Bian<sup>2</sup> · Junqing Li<sup>2</sup>

Received: 24 March 2021 / Accepted: 19 October 2022 / Published online: 29 November 2022  
© The Author(s) 2022

## Abstract

Many industrial applications require time-consuming and resource-intensive evaluations of suitable solutions within very limited time frames. Therefore, many surrogate-assisted evaluation algorithms (SAEAs) have been widely used to optimize expensive problems. However, due to the curse of dimensionality and its implications, scaling SAEAs to high-dimensional expensive problems is still challenging. This paper proposes a variable surrogate model-based particle swarm optimization (called VSMPSO) to meet this challenge and extends it to solve 200-dimensional problems. Specifically, a single surrogate model constructed by simple random sampling is taken to explore different promising areas in different iterations. Moreover, a variable model management strategy is used to better utilize the current global model and accelerate the convergence rate of the optimizer. In addition, the strategy can be applied to any SAEA irrespective of the surrogate model used. To control the trade-off between optimization results and optimization time consumption of SAEAs, we consider fitness value and running time as a bi-objective problem. Applying the proposed approach to a benchmark test suite of dimensions ranging from 30 to 200 and comparisons with four state-of-the-art algorithms show that the proposed VSMPSO achieves high-quality solutions and computational efficiency for high-dimensional problems.

**Keywords** Expensive optimization · Fitness estimation · Simple random sampling · Model management strategy

**Arabic Keywords** (البدنية اللياقة تقدير) · (التمن باهظة التحسين مشكلة): الرئيسي الكلمات · (النموذجية الإدارة استراتيجية) · (بسيطة عشوائية عينات أخذ)

## Introduction

Compared to conventional optimization algorithms, evolutionary algorithms (EAs) are more apt at handling a number

of complex problems in real-world applications [1,2]. EAs have, therefore, been widely applied in many real-world applications, including drug design [3], control engineering applications [4], and wing configuration design [5]. However, EAs generally require thousands of fitness evaluations to achieve a satisfactory candidate solution. In many engineering optimization computations, a single numerical simulation can take several minutes, hours, or even days to complete. Examples of this include computational fluid dynamics (CFD) simulation, in which performing a single simulation to evaluate a candidate design generally requires several hours. Furthermore, the number of required additional fitness evaluations rises with the dimension of the optimization problem, resulting in high computational costs to run hundreds or thousands of fitness evaluations. To solve such expensive optimization problems, surrogate model-based EAs, in which a surrogate model (also called a meta-model) is applied instead of the expensive original function, are often used.

---

✉ Jie Tian  
tianjie1023@outlook.com

Mingdong Hou  
houmingdong@sdtju.edu.cn

Hongli Bian  
2020317062@stu.sdn.edu.cn

Junqing Li  
lijunqing@lcu-cs.com

- <sup>1</sup> School of Data and Computer Science, Shandong Women's University, Jinan, China
- <sup>2</sup> School of Information Science and Engineering, Shandong Normal University, Jinan, China
- <sup>3</sup> School of Information Science and Electrical Engineering, Shandong Jiaotong University, Jinan, China

Over the last few decades, a variety of surrogate-assisted evolutionary algorithms (SAEAs) have been identified. Existing strategies for SAEAs to employ surrogate models can generally be divided into the single-surrogate and multi-surrogate model-based strategies, depending on the number of surrogate models that SAEAs have used. Although many generic machine learning methods have been used to build surrogate models, no specific rule has been proposed to determine the type of model most suitable for use as a surrogate [6]. In general, single-surrogate model-based EAs employ the Gaussian process (GP) model, most likely because this model can predict a candidate solution while providing an estimate of the error of the predicted value. There are several infill sampling criteria that apply GP-provided prediction and error estimation approaches, including the expected improvement infill criterion [7,8] and lower confidence bound infill criterion [9,10], have been proposed as guides for achieving promising solutions. Most non-GP models, including polynomial regression surface models [11,12], artificial neural networks [13,14], radial basis functions (RBFs) [15–17] and many others [18], can only provide predictions and cannot provide error estimates for their predictions. Because of these limitations, multiple model-based EAs involving the production of multiple predictions from multiple models are applied to avoid cases in which the algorithms run into local optima.

For SAEAs, the core problem is how to use surrogate models to guide the optimization process reasonably. When an expensive problem has a high-dimensional decision space, it becomes more challenging for SAEAs to employ surrogate models to guide the optimization process effectively. Firstly, the number of training samples required by the surrogate model grows exponentially as the problem dimensions increase [19]. This implies more experimental evaluations are required, often expensive and infeasible in real applications. Due to the lack of samples on high-dimensional expensive problems, it is difficult to construct a single surrogate model with high accuracy [20]. It is commonly known that the use of inaccurate surrogates might cause an optimization process to be misleading. Second, it needs more time to create a surrogate model as the problem dimensions increase. For example, in the Gaussian process (GP) model, global optimization for high-dimensional acquisition functions is intrinsically a hard problem and can be prohibitively expensive to be feasible [21]. Generally speaking, existing research on extending the SAEAs to high-dimensional expensive problems can be roughly classified into three categories:

The first strategy is to deal with the lack of samples through data processing and data generation methods. DDEA-PES [22] used data perturbation to generate diverse datasets. SAE0 [23] trained and activated the surrogate model only after enough data samples were collected. ESAO [24] randomly projected training samples into a set of low-

dimensional sub-spaces rather than training in the original high-dimensional space.

The second strategy is to improve the performance of surrogate models. In [25], a GP model was combined with the partial least squares method to solve high-dimensional problems with up to 50 design variables. In our previous work, we proposed a multi-objective infill criterion [26] for GP model management. TR-SADEA [27] employs a self-adaptive GP model for antenna design. The RBF-assisted approach based on granulation was proposed in [28,29]. In [30], a radial basis function network (RBFN) with a trust-region approach was considered a local model for solving 20-dimensional problems. Wang and Jin [31] employed three widely used models (i.e., PR, RBF, and GP) to construct both one global ensemble model and a local ensemble model respectively. Li et al. [32] employed two criteria to balance exploitation and convergence to solve medium-scaled computationally expensive problems. MS-RV [33] transferred the knowledge from the coarse surrogate to the fine surrogate in off-line data-driven optimization.

The third strategy is to improve optimization efficiency by multiple swarms. Multiple swarms were used in SA-COSO [34] for solving high-dimensional problems ranging from 30 to 200 dimensions. Pan et al. [35] proposed an efficient surrogate-assisted hybrid optimization (SAHO) algorithm that combines two EAs (TLBO and DE) as the basic optimizer for 100-dimensional problems.

This paper proposes a variable surrogate model-based particle swarm optimization (VSMPSO) algorithm for high-dimensional expensive problems. To the best of our knowledge, VSMPSO is the first attempt to extend a single surrogate-assisted EA to solve the 200 dimension problems. The main contributions of this paper are as follows:

- The proposed VSMPSO, does not focus on improving the accuracy of surrogate models, but rather relies on the blessing of uncertainty [36], which only employs one RBF model as a single surrogate in combination with the proposed variable surrogate model strategy to explore different promising area in different generation to avoid model misdirection throughout the whole optimization process.
- The prediction ability of the surrogate model is not only used to predict the current population. For deep mining of the surrogate model prediction information, the most promising point of the surrogate model would be found and transferred into the optimizer population to accelerate the optimization.
- The proposed algorithm framework of VSMPSO can be applied in any surrogate-assisted evolutionary algorithm irrespective of the surrogate model used.

The remainder of this paper is organized as follows: the next section introduces a brief overview of the related techniques used in this paper. The main framework of the proposed algorithm is then presented in the subsequent section. The penultimate section compares a few state-of-the-art algorithms with widely used benchmark problems with 30, 50, 100, and 200 dimensions. The final section provides the conclusion.

## Related techniques in VSMPSO

### Particle swarm optimization (PSO)

The canonical PSO, developed by Eberhart and Kennedy in 1995, is a population- or swarm-based intelligent optimisation algorithm inspired by the social behaviours of populations of organisms such as birds (flocking) or fish (schooling) [37]. Eq. (1) and Eq. (2) describe the evolution of  $x_j$  (the position of the  $j$ th individual at generation  $(t + 1)$ ) along the  $d$ th dimension in the canonical PSO:

$$x_j^d(t + 1) = x_j^d(t) + \Delta x_j^d(t + 1) \tag{1}$$

$$\begin{aligned} \Delta x_j^d(t + 1) = & \omega \Delta x_j^d(t) + c_1 r_1 \cdot (Pbest_j^d(t) - x_j^d(t)) \\ & + c_2 r_2 \cdot (Gbest^d(t) - x_j^d(t)). \end{aligned} \tag{2}$$

The feature distinguishing the canonical PSO from other EAs such as the genetic algorithm (GA) or differential evolution (DE) is that it converges rapidly but easily falls into local optima. To prevent premature convergence, a variety of modified PSOs have been proposed, including the comprehensive learning PSO [38], distance-based locally informed PSO [39], social learning PSO [40], and competitive swarm optimiser (CSO) [41]. Based on the effective performance of social learning particle swarm optimization (SLPSO), we propose a simplified SLPSO to generate candidate solutions whose primary structure is similar to the SLPSO algorithm proposed by Cheng and Jin. In this simplified SLPSO, individual  $x_j$  are updated using the following formulas:

$$x_j^d(t + 1) = \begin{cases} x_j^d(t) + \Delta x_j^d(t + 1) & \text{if } p_j(t) \leq P_j^L \\ x_j^d(t) & \text{otherwise} \end{cases} \tag{3}$$

$$\begin{aligned} \Delta x_j^d(t + 1) = & r_1 \cdot \Delta x_j^d(t) + r_2 \cdot (x_k^d(t) - x_j^d(t)) \\ & + r_3 \cdot \varepsilon \cdot (\bar{x}_d(t) - x_j^d(t)), \end{aligned} \tag{4}$$

where  $1 \leq j < N$ ,  $N$  is the population size,  $1 \leq d \leq D$ , and  $D$  is the dimension of the search space. In each generation, the population is sorted according to fitness value from bad to good, with  $x_1$  and  $x_N$  representing the worst and best

solutions, respectively, at the current generation.  $x_k$  is a randomly chosen demonstrator for  $x_j$ ,  $j < k \leq N$ , and  $x_k^d(t)$  represents the  $d$ th element of  $x_k$ . We note that a demonstrator should be chosen for each element of  $x_j$ .  $P_j^L$  is the learning probability, which is inversely proportional to the fitness of  $x_j$ ,  $p_j(t)$  is a randomly generated probability for  $x_j$ ,  $r_1$ ,  $r_2$ , and  $r_3$  are random numbers in the range  $[0, 1]$ , and  $\varepsilon$  is the social influence factor that controls the influence of  $\bar{x}_d(t)$ . In Eq. (4),  $\bar{x}_d(t)$  is the mean position along the  $d$ th dimension of the population at generation  $t$ . If a uniform sampling method such as Latin hypercube sampling (LHS) is used in the initialisation process,  $\bar{x}_d(t)$  can be quite close to the  $1 \times D$  zero vector  $o = [0, \dots, 0]$ . The function of the global optimum at the zero vector  $o$  can easily lead the population toward a promising region, and to avoid this coincidence, we set the parameter  $r_3$  to zero to remove the effect of  $\bar{x}_d(t)$ . This simplifies Eq. (4) to

$$\Delta x_j^d(t + 1) = r_1 \cdot \Delta x_j^d(t) + r_2 \cdot (x_k^d(t) - x_j^d(t)) \tag{5}$$

In this study, we generated new swarms using Eq. (3) and Eq. (5), and in the following sections, the variant SLPSO is referred to as ‘PSO’ for brevity.

### RBF network

The functionality of an RBF network as a type of neural network was described in detail in [42] and can be represented by the following form:

$$\hat{f}(x) = \sum_{i=1}^M \omega_i \phi(\|x - x_i\|), \tag{6}$$

where  $x \in \mathbb{R}^D$  is an input vector,  $\phi$  is the basic function of the RBF network,  $\|\cdot\|$  is the 2-norm (also called the Euclidean norm),  $\omega_i$  is the weight vector,  $M$  represents both the number of input units in the RBF input layer and the number of samples for building the RBF model. Because the basic function  $\omega_i$  is one of the key factors affecting the performance of the model, many forms of  $\omega_i$  have been developed, including multi-quadric, thin plate spline, Gaussian, and cubic forms. A comparison of different choices of  $\omega_i$  in [43] revealed that the thin plate spline and linear and cubic RBFs theoretically perform better than either the multi-quadric or Gaussian RBF. Additionally, numerical investigation results have demonstrated that the cubic RBF can improve the performance of the thin plate spline and multi-quadric RBFs [44]. Furthermore, cubic RBF-assisted EAs have been successfully used in local function approximation. Based on these previous studies, the proposed method employs cubic basic function to construct RBF network, which is a common machine learning technique for fitness approximation [2,8].

The basic function of the cubic RBF employed in this study is  $\phi(\|x - x_i\|) = (\|x - x_i\|)^3$ .

## Proposed VSMPSO algorithm

### VSMPSO framework

The main framework shown in Fig. 1 presents the overall algorithm for VSMPSO. The solid black arrow lines represent the flow direction of the algorithm, and the green dot arrow dotted line to mark the data flow direction. At the beginning of VSMPSO, the initial individuals are generated, which are all evaluated and then taken into the database **DB**. In each generation, the variable model management strategy was proposed to decide how to construct the surrogate model for fitness estimation and select promising solutions for fitness evaluation. In the variable model management strategy (Part I) of Fig. 1, the simple random sampling method is used to select samples from the database **DB** for building an RBF model. Subsequently, the RBF model will be carried out to find the most promising point in the global search space. This is followed by the knowledge transferred from the RBF model to the current population. Then, in the variable model management strategy (Part II) of Fig. 1, the infill criterion used in VSMPSO considers potential optimum points to be evaluated and put into **DB**.

The details for the main components in VSMPSO have been explained in Algorithm 1. The optimizer used in VSMPSO is a variant of that used in SLPSO. Steps 2–4 of Algorithm 1 describe the steps for generating the initial population (called  $P(1)$ ) by Latin Hypercube Sampling (LHS) and then creating the database **DB** by all the initial individuals in  $P(1)$  with their real fitness values. As shown in steps 6–11, the main optimization loop contains two main components of the proposed algorithm that will be described in detail in Algorithms 2 and 3. In steps 8 and 10, the variable model management strategy is proposed to design model construction and the infill criterion, which will be described in “[Variable model management strategy](#)” section. In step 9, the knowledge from the RBF model will be transferred to  $P(t)$  (the population in generation  $t$ ). Finally, the program outputs the most satisfying solution with its real fitness value and end. Overall speaking, VSMPSO contains two-layer loops. The outer loop uses the variant SLPSO as the optimizer in Algorithm 1, and the inner loop uses the canonical PSO as the optimizer in Algorithm 3. In Algorithm 3, the current RBF model is used as the objective function to drive the canonical PSO iterations, and then, the global best solution found by canonical PSO is transferred to variant SLPSO. In the following subsections, we will detail the two main components of the proposed algorithm, i.e., the variable model management

strategy and the knowledge transfer from model to population.

---

### Algorithm 1: The pseudo-code of VSMPSO

---

```

Output : the global best solution with its real fitness
1 begin
2    $t = 1$ ;
3   Use LHS to initialize the population  $P(1)$  and evaluate the
   swarm by the original expensive function;
4   Save  $P(1)$  with their real fitness in DB;
5   while computational budget is not reaching the upper limit do
6      $t = t + 1$ ;
7     Update the individuals of  $P(t)$  according to Eq.(3) and
   Eq.(5);
8     Building RBF model and Fitness estimation by steps 2–4
   of Algorithm 2;
9     Transfer the knowledge from the RBF model to the
   current population  $P(t)$  by Algorithm 3;
10    Select two potential optimum points to be evaluated and
   update DB by steps 5–7 of Algorithm 2;
11    Update the global best solution;
12  end while
13 end

```

---

### Variable model management strategy

The key issues influencing the performance of surrogates are mainly model selection and model management. First, for model selection, according to the previous work, on high-dimensional problems, constructing GP models becomes time-consuming, but RBF has been proven to perform better with small samples than other common surrogate models [45], so we determine RBF as a surrogate model. Additionally, for model management, as mentioned in the “[Introduction](#)” section, due to the lack of samples on high-dimensional expensive problems, it is not easy to construct a single surrogate model with high accuracy. Since the blessing of uncertainty and the multiple local optima of original expensive problems, an accurate surrogate model is not always necessarily in optimization. So in this work, unlike the previous work focusing on improving the accuracy of models, only one global surrogate model is trained. Furthermore, due to different samples that may construct models toward different promising areas, a global model management strategy inspired by simple random sampling is proposed to enhance the diversity of the single model between each generation and thus avoid persistent misleading in a wrong direction throughout the whole optimization process.

As observed from the pseudo-code of Algorithm 1, the RBF model is updated during each generation, in step 2 of Algorithm 2,  $\lambda = [M \times 80\%]$  samples are selected using simple random sampling, with the number of selected samples  $\lambda$  accounting for 80% of the total sample size based on a



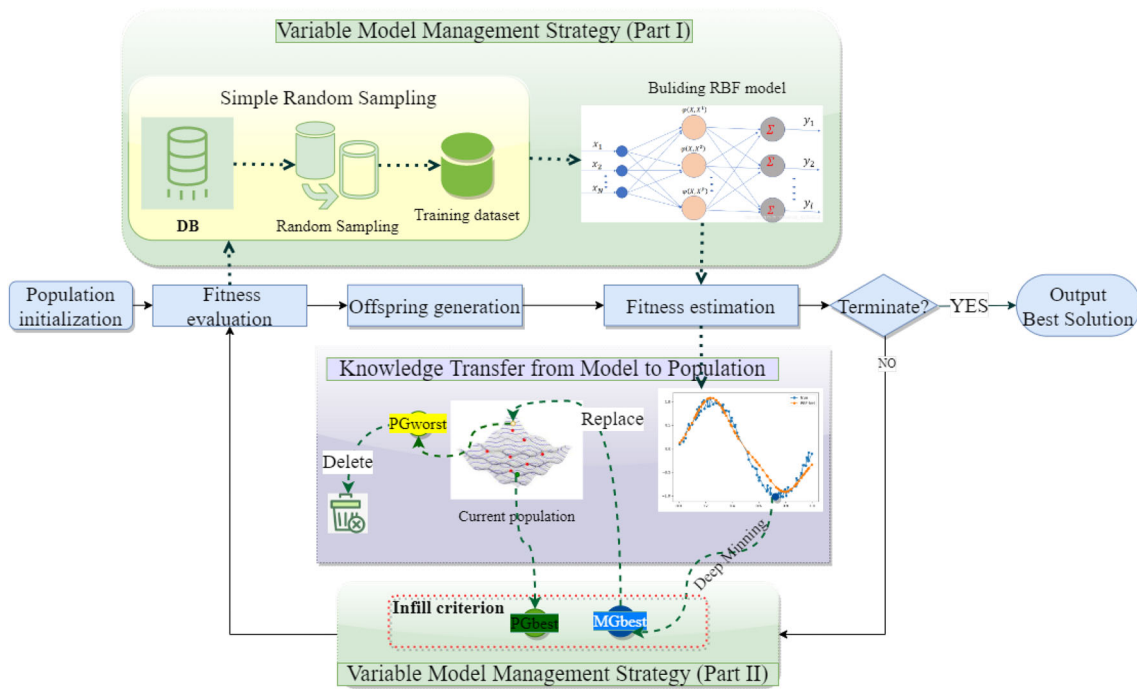


Fig. 1 VSMPSO flow diagram

**Algorithm 2:** The pseudo-code of Variable Model Management Strategy

**Require:** *DB* (Database with all solutions evaluated by the exact original function),  $P(t)$ , *MGbest*  
**Output :** *PGbest* and *MGbest* with their real fitness, *PGworst*, Current RBF model

- 1 begin
- 2 Select random  $\lambda$  solutions from *DB* using a simple random sampling strategy;
- 3 Train the RBF model ;
- 4 Estimate  $P(t)$  using RBF model;
- 5 Sort  $P(t)$  and obtain *PGbest* and *PGworst*;
- 6 Reevaluate *PGbest* and *MGbest* by the original expensive function;
- 7 Place the two solutions (*PGbest* and *MGbest*) along with their real fitness value into *DB*;
- 8 end

common empirical value used in  $K$ -fold cross-validation [46] with  $K = 5$ . Furthermore, the effectiveness of this strategy and sample size will be further verified in “Effects of variable model management strategy” and “Parameter sensitivity analysis”. After estimating the fitness value of  $P(t)$  by the current RBF model in step 4, the current population  $P(t)$  is sorted according to the estimated fitness value from bad to good in step 5, with  $x_1$  and  $x_N$  representing the worst solution *PGworst* and the best solution *PGbest*, respectively. As shown in steps 6–7, two potential optimum points are considered to be evaluated by the original expensive function. The first is *PGbest*, (the global optimum individual of the

current population), and the second is *MGbest*, (the most promising point of the current surrogate model, obtained from Algorithm 3).

**Knowledge transfer strategy**

As mentioned in Algorithm 2, in each iteration of the optimization loop, a simple random sampling method selects different samples to construct the RBF model, and those RBF models may contribute toward different promising areas in different generations. So in Step 3 of Algorithm 3, for further data mining of the RBF model, the RBF model is considered as an objective function, which is defined as follows:

$$\begin{aligned} \min \quad & F_{\text{RBF}}(\mathbf{x}) \\ \text{s. t.} \quad & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u, \end{aligned} \tag{7}$$

where  $\mathbf{x} \in \mathbb{R}^D$  is the feasible solution set, denotes the same search space as original expensive problem,  $F_{\text{RBF}}(\mathbf{x})$  is the objective function,  $\mathbf{x}_l$  and  $\mathbf{x}_u$  are the lower and upper bounds of the decision variables. In steps 4–9, a canonical PSO is employed to find the global optimum of the current RBF model. In step 6, the canonical PSO updates individuals using Eq. (1) and (2). In step 8, *Pbest* (the personal best of the particle) and *Gbest* (the global best of the swarm) are updated in each generation. After the iteration is completed, in step 11, *Gbest*, the final best solution of the RBF model obtained through the canonical PSO, is assigned *MGbest*, which is considered the most promising point of the RBF model. In

the following steps, by substituting  $MGbest$  for  $PGworst$  in  $P(t)$ , the knowledge transferred from the RBF model to the population is expected to be able to enhance the search performance of VSMPSO by reducing the likelihood of getting stuck in a local optimum. Furthermore, the effectiveness of the knowledge transfer strategy is further verified in “Effects of variable model management strategy” section by comparing VSMPSO with SMPSO.

---

**Algorithm 3:** The pseudo-code of Knowledge Transfer
 

---

**Require:**  $PGworst$ ,  $P(t)$  (the current population in generation  $t$ ); Current RBF model

**Output :**  $P(t)$ ,  $MGbest$  with its estimated fitness value

```

1 begin
2   /*Data mining from the current RBF model in each
   generation of Algorithm 1 */
3   Consider Eq.(7) as the objective function for data mining of
   RBF model;
4   Initialize the population and evaluate them using the RBF
   model;
5   while computational budget is not reaching the upper limit do
6     Update the population using Eq. (1) and (2).
7     Evaluate the fitness of all individuals using the RBF
   model;
8     Update  $Pbest$  and  $Gbest$ ;
9   end while
10   $MGbest = Gbest$ ;
11   $P(t).remove(PGworst)$ ;
12   $P(t).add(MGbest)$ ;
13 end

```

---

## Experimental studies

A series of empirical studies on eleven commonly-used benchmark functions (for details, see attached Table 1) are designed to verify the effectiveness and optimality of the proposed VSMPSO. These eleven functions have different characteristics, including unimodal, multimodal, and very complex multimodal functions. Most of them are very difficult to optimize. F1–F7 functions are commonly used by other SAEAs [26,31]. F8–F11 functions are adopted from test suite CEC 2017 [48], which have been recently proposed and are comparatively complex. The dimensions of these eleven functions are tested from 10 to 200. The CEC 2017 functions are only tested with dimension 100 because the highest dimension of the functions in this test suite is 100. The statistical results of the compared algorithms are given on 30-D, 50-D, 100-D, and 200-D problems, respectively. Furthermore, the best results are highlighted in bold. The proposed VSMPSO is compared with several popular and recently proposed SAEAs, SAHO [35], SACOSO [34], CAL-SAPSO [31] and MGP-SLPSO [26] under different dimensions. Moreover, we consider the best fitness value

and the running time as the bi-objective problem to verify the performance of VSMPSO.

## Experimental setup in details

All experiments are implemented on a high computing capability server with an Intel XEON E5-2620v4 processor with 64GB in RAM. Each algorithm was run 30 times in Matlab 2020B to eliminate the effect of statistical variation. We applied Friedman’s test to determine if there are any significant differences in the best fitness values obtained by the algorithms. The runs were performed using MATLAB’s Statistics Toolbox. The  $p$  values were obtained using Friedman’s test on pairwise comparisons between VSMPSO and other comparison algorithms. Normally, the  $p$  value threshold for statistical significance is 0.05 [49], and  $p \geq 0.05$  showed no significant difference; whereas  $p < 0.05$  showed significant difference. “+” indicates the labeled algorithm with the mean best value significantly outperformed other algorithms. The RBF model used in VSMPSO and other algorithms was implemented using the MATSuMoTo Toolbox [50]. The parameters of the optimizer SLPSO were set as recommended in [40]. For all the compared algorithms in this paper, the termination condition depended on the number of consumed function evaluations (FEs). The computational budget was less than  $11 \cdot D$  number of function evaluations (NFEs) [31,32], which means the limited number of fitness evaluations was 11 times the dimension of the problem.

We describe the algorithm performance through two chart types to show convergence profiles and distribution of solutions, such as Figs. 2 and 3. It is noted that the original author provides the source code of SA-COSO and SAHO and the source codes of MGP-SLPSO and CAL-SAPSO are available on the internet. In [31], CAL-SLPSO is examined on problems less or equal to 30-D. So we compare statistical results with MGP-SLPSO, CAL-SAPSO, SAHO, and SACOSO on 30-D problems and then compare statistical results with MGP-SLPSO, SAHO, and SACOSO on 50- and 100-D problems. Considering the time consumption, we compare statistical results with MGP-SLPSO and SACOSO on 200-D problems.

## Numerical results on 30- and 50-D F1–F7 functions

The proposed VSMPSO can be applied in any surrogate-assisted evolutionary algorithm irrespective of the surrogate model used. In this paper, we used two widely adopted models, GP and RBF, to predict fitness values, referred to as VSMPSO(GP) and VSMPSO(RBF). VSMPSO(GP) used GP as a surrogate model, while VSMPSO(RBF) used RBF. Both VSMPSO(GP) and VSMPSO(RBF) are all based on the VSMPSO framework in Algorithms 1, 2 and 3; their only difference was the different surrogate models used.

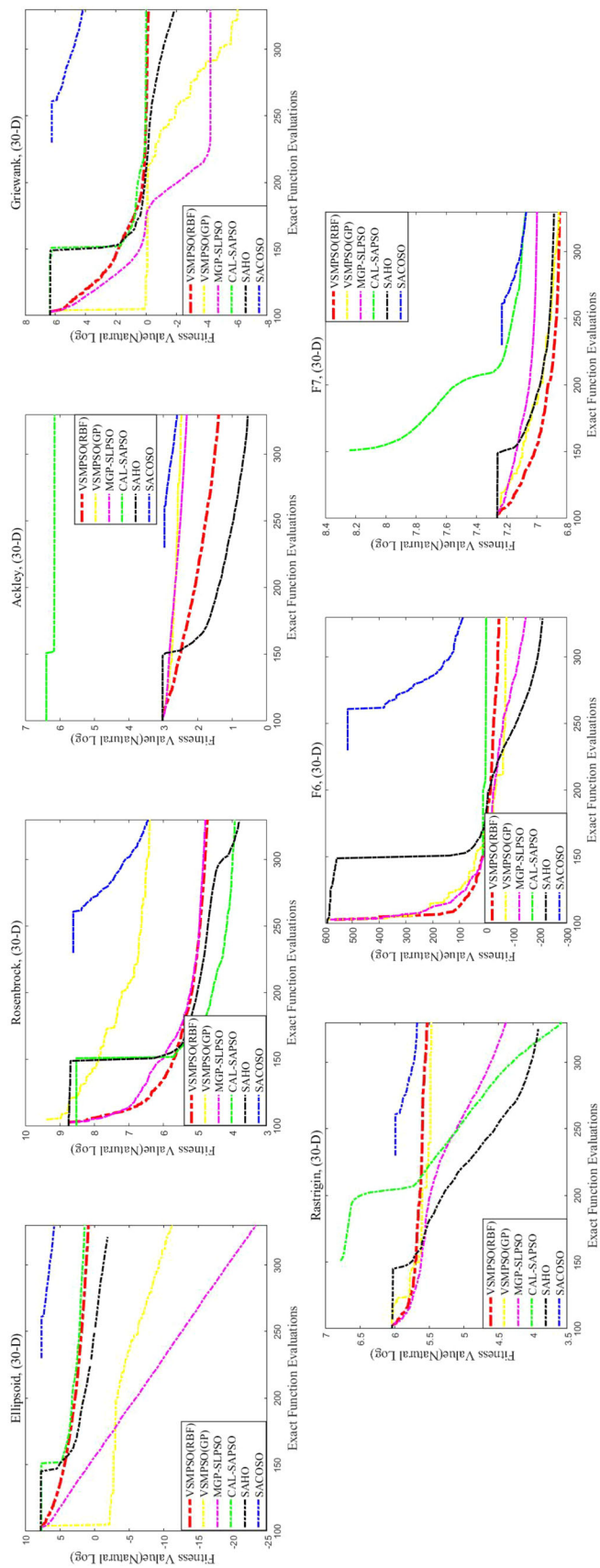


Fig. 2 Convergence profiles of MGP-SLPSO, CAL-SAPSO, SAHO, SACOSO, VSMPSO(GP) and VSMPSO(RBF) on 30D F1–F7 with 330 FEs

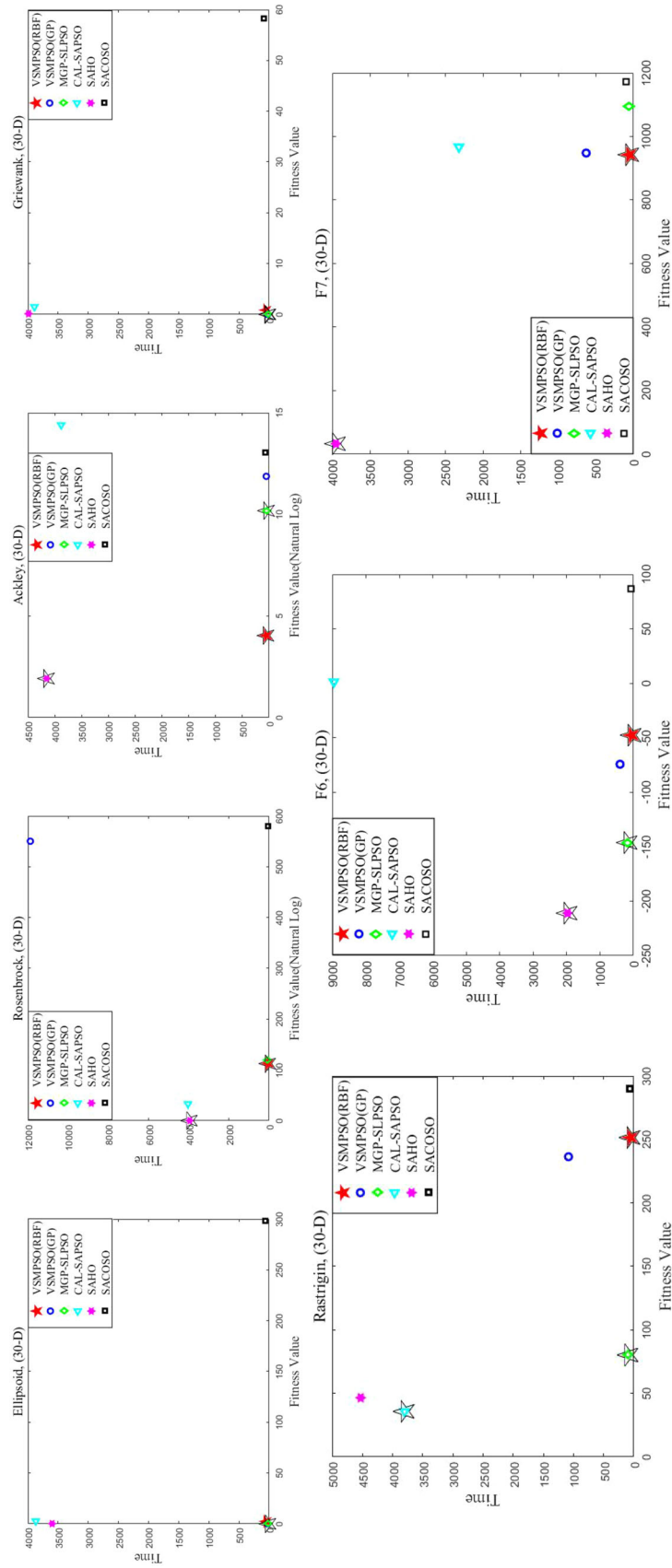


Fig. 3 Distribution of solutions of MGP-SLPSO, CAL-SAPSO, SAHO, SACOSO and VSMPSO on 30D F1–F7 with 330 FEs



**Table 1** Description of benchmark functions

Function	No. of variables	Global optimum
F1: Ellipsoid	30/50/100/200	0
F2: Rosenbrock		0
F3: Ackley		0
F4: Griewank		0
F5: Rastrigin		0
F6: Shifted rotated Rastrigin function (F10 in [47])		− 300
F7: Rotated hybrid composition function (F19 in [47])		10
F8: Shifted rotated Rastrigin function (F25 in [48])	50/100	2500
F9: Rotated hybrid composition function (F26 in [48])		2600
F10: Rotated hybrid composition function (F27 in [48])		2700
F11: Rotated hybrid composition function (F28 in [48])		2800

From Table 2, MGP-SLPSO was the best on F1, VSMPSO (GP) the best on F4, VSMPSO(GP) on F7, CAL-SAPSO on F5. SAHO was the best on F2, F3, and F6. So SAHO had better optimization results than other SAEAs; however, from Table 3, SAHO and CAL-SAPSO took hundreds of times longer than other SAEAs. We considered optimization results and the running time as the bi-objective problem. In the first row of Table 3, “Best fitness” means the average value of the best fitness values obtained by 30 independent operations performed by each algorithm, and “Time” means the mean time obtained by 30 independent operations performed by each algorithm. We consider “Best fitness” and “Time” as two objectives, referred to as the non-dominated solution (NDS) and dominated solution (DS). We sorted the comparison algorithm results and marked all non-dominated solutions (NDSs) on the Pareto front with a black hollow pentagram in Fig. 3. From Table 3, we can see that VSMPSO (RBF) achieved NDS on F1, F4, F5, F6 and F7, similar to MGP-SLPSO. From Fig. 3, on F1 and F3, the dominant solution (DS) achieved by VSMPSO (RBF) is the closest to NDS. In Fig. 2, the convergence rate of VSMPSO (RBF) is second only to SAHO on F3 and has the fastest convergence curve on F7. MGP-SLPSO had the shortest running time because Matérn32 function was used as the kernel function. In our previous work [51], when the commonly used squared exponential (SE) kernel function is used, compared with using Matérn32, the running time of GP model-assisted SLPSO even increases over ten times. SE is the most commonly adopted covariance function in GP-assisted optimization algorithms, for example, in [9,15,52,53].

We compared VSMPSO with MGP-SLPSO, SAHO, and SACOSO on 50D functions. In Table 5, we can observe that MGP-SLPSO achieved the best mean value on F1 and F4, and SAHO achieved the best mean value on F2, F3, F5, F6, and F7. VSMPSO(RBF) achieved a similar performance with SAHO on F7. From Table 3 and Fig. 5, VSMPSO(RBF) derived NDS on F3, F5, F6, and F7, which is one less than

MGP-SLPSO and SAHO. From Fig. 4, the convergence curve of VSMPSO(RBF) is very close to the SAHO one on F6 and F7. From Fig. 5, the dominant solution(DS) achieved by VSMPSO(RBF) is the closest to NDS on F1, F2, and F4. In Table 4, the running time spent by VSMPSO(GP) is dozens or even hundreds of times that of VSMPSO(RBF); the running time spent by VSMPSO(GP) is dozens of times higher than that of VSMPSO(RBF) on F1, F4, and F6 functions, and more than 100 times higher on F2, F3, F5, and F7. The slow computational efficiency makes VSMPSO(GP) unable to run on higher dimensions. Furthermore, from Table 4 and Fig. 5, VSMPSO(GP) has no NDS on 50D functions, whereas VSMPSO(RBF) has 4 NDSs. The overall performance of VSMPSO(RBF) is better than VSMPSO(GP). Hence, in the next comparative test, we used VSMPSO(RBF) algorithm to compare with other algorithms; it is abbreviated as VSMPSO in the following algorithm comparison.

### Numerical results and analysis on 50D F1–F7 functions with 2000 FEs

To further observe the optimization ability of VSMPSO, we extended the computational budget to 2000 FEs. From the aforementioned analysis, VSMPSO (GP) was time-consuming, and its optimal performance was inferior to VSMPSO(RBF). Therefore, in the following algorithm comparison, we only used VSMPSO (RBF), referred to as VSMPSO. Although VSMPSO did not achieve good optimization performance on the 50D with 1000 FEs, it has better performance with 2000 FEs. In Table 6, MGP-SLPSO achieved the best mean value on F1; SAHO achieved the best mean value on F2, F3, F4, and F7. VSMPSO achieved the best mean value on F5 and F6, and has similar performance and no significant difference with SAHO on F7. Furthermore, VSMPSO achieved the second-best mean value behind SAHO on F2 and F3, and the running time of VSMPSO was smaller than SAHO. MGP-SLPSO algorithm took the

**Table 2** Statistical results of MGP-SLPSO, CAL-SAPSO, SAHO, SACOSO and VSMPSO on 30D F1–F7 with 330 FEs

Function	Algorithm	Best	Worst	Mean (Friedman test)	SD
F1: Ellipsoid	VSMPSO(GP)	7.88E−06	1.68E−05	1.34E−05	2.17E−06
	VSMPSO(RBF)	2.67E−01	9.20E+00	2.33E+00	1.81E+00
	MGP-SLPSO	<b>2.31E−11</b>	<b>1.41E−10</b>	<b>7.22E−11(+)</b>	<b>3.05E−11</b>
	CAL-SAPSO	2.64E−01	1.14E+01	2.69E+00	2.68E+00
	SAHO	1.33E−02	4.15E−01	1.39E−01	1.03E−01
	SACOSO	1.59E+02	4.98E+02	2.98E+02	8.36E+01
F2: Rosenbrock	VSMPSO(GP)	5.53E+01	7.62E+01	7.40E+01	6.59E+00
	VSMPSO(RBF)	1.03E+02	2.13E+02	1.52E+02	2.80E+01
	MGP-SLPSO	7.28E+01	2.87E+02	1.20E+02	3.91E+01
	CAL-SAPSO	3.51E+01	9.64E+01	5.34E+01	<b>1.29E+01</b>
	SAHO	<b>2.93E+01</b>	<b>9.58E+01</b>	<b>3.30E+01(+)</b>	2.09E+01
	SACOSO	2.71E+02	1.09E+03	5.80E+02	2.00E+02
F3: Ackley	VSMPSO(GP)	4.16E+00	1.34E+01	1.19E+01	3.36E+00
	VSMPSO(RBF)	3.43E+00	5.15E+00	4.03E+00	<b>4.68E−01</b>
	MGP-SLPSO	5.53E+00	1.77E+01	1.02E+01	3.10E+00
	CAL-SAPSO	8.00E+00	1.96E+01	1.44E+01	3.31E+00
	SAHO	<b>6.54E−02</b>	<b>2.54E+00</b>	<b>1.92E+00(+)</b>	5.83E−01
	SACOSO	1.09E+01	1.65E+01	1.30E+01	1.36E+00
F4: Griewank	VSMPSO(GP)	<b>2.15E−03</b>	<b>3.35E−03</b>	<b>2.39E−03(+)</b>	<b>4.99E−04</b>
	VSMPSO(RBF)	6.50E−01	1.00E+00	8.54E−01	8.07E−02
	MGP-SLPSO	6.70E−03	3.44E−02	1.43E−02	6.20E−03
	CAL-SAPSO	1.22E+00	1.88E+00	1.49E+00	1.49E−01
	SAHO	1.01E−02	4.55E−01	1.40E−01	1.12E−01
	SACOSO	3.74E+01	8.99E+01	5.83E+01	1.37E+01
F5: Rastrigin	VSMPSO(GP)	2.12E+02	2.64E+02	2.36E+02	2.02E+01
	VSMPSO(RBF)	2.01E+02	2.94E+02	2.51E+02	2.42E+01
	MGP-SLPSO	2.90E+01	1.46E+02	8.02E+01	2.82E+01
	CAL-SAPSO	1.97E+01	<b>5.84E+01</b>	<b>3.58E+01(+)</b>	<b>1.20E+01</b>
	SAHO	<b>5.04E+00</b>	1.12E+02	4.66E+01	2.70E+01
	SACOSO	2.35E+02	3.34E+02	2.90E+02	2.21E+01
F6 (F10 in CEC05)	VSMPSO(GP)	− 8.58E+01	− 5.25E+01	− 7.43E+01	1.27E+01
	VSMPSO(RBF)	− 8.90E+01	1.26E+01	− 4.76E+01	2.54E+01
	MGP-SLPSO	− 2.12E+02	− 5.39E+01	− 1.46E+02	3.41E+01
	CAL-SAPSO	2.49E−01	2.20E+00	1.32E+00	<b>5.83E−01</b>
	SAHO	− <b>2.68E+02</b>	− <b>1.21E+02</b>	− <b>2.11E+02(+)</b>	3.34E+01
	SACOSO	− 4.51E+00	1.45E+02	8.67E+01	3.77E+01
F7 (F19 in CEC05)	VSMPSO(GP)	9.40E+02	<b>9.56E+02</b>	9.48E+02(≈)	<b>6.13E+00</b>
	VSMPSO(RBF)	<b>9.26E+02</b>	9.97E+02	<b>9.43E+02(+)</b>	1.51E+01
	MGP-SLPSO	1.01E+03	1.18E+03	1.10E+03	3.65E+01
	CAL-SAPSO	1.06E+03	1.34E+03	1.18E+03	5.54E+01
	SAHO	9.29E+02	1.10E+03	9.67E+02	3.60E+01
	SACOSO	1.07E+03	1.28E+03	1.17E+03	5.07E+01

**Table 3** Statistical results (best fitness and cost time) obtained by MGP-SLPSO, CAL-SAPSO, SAHO, SACOSO, VSMPSO(GP) and VSMPSO(RBF) on 30D F1–F7 with 330 FEs

Fuction 30D/330FEs	VSMPSO(GP) Best fitness/time	VSMPSO(RBF) Best fitness/time	MGP-SLPSO Best fitness/time	CAL-SAPSO Best fitness/time	SAHO Best fitness/time	SACOSO Best fitness/time
F1: Ellipsoid	1.34E - 05/2.37E+01	2.33E+00/5.45E+01	7.22E - 11/1.45E+01	2.69E+00/3.97E+03	1.39E - 01/3.60E+03	2.98E+02/6.07E+01
F2: Rosenbrock	7.40E+01/2.93E+01	1.52E+02/6.86E+00	1.20E+02/8.15E+01	5.34E+01/4.04E+03	3.30E+01/3.95E+03	5.80E+02/6.86E+01
F3: Ackley	1.19E+01/5.34E+01	4.03E+00/5.52E+01	1.02E+01/4.63E+01	1.44E+01/3.88E+03	1.92E+00/4.15E+03	1.30E+01/7.00E+01
F4: Griewank	2.39E - 03/2.47E+01	8.54E - 01/5.64E+01	1.43E - 02/1.43E+01	1.49E+00/3.87E+03	1.40E - 01/3.99E+03	5.83E+01/8.43E+01
F5: Rastrigin	2.36E + 02/1.08E + 03	2.51E + 02/5.52E + 01	8.02E + 01/8.79E + 01	3.58E + 01/3.80E + 03	4.66E + 01/4.53E + 03	2.90E + 02/6.48E + 01
F6 (F10 in CEC05)	- 7.43E + 01/4.02E+02	- 4.76E + 01/5.55E + 01	- 1.46E + 02/1.72E + 02	1.32E + 00/8.95E + 03	- 2.11E + 02/1.97E + 03	8.67E + 01/8.08E + 01
F7 (F19 in CEC05)	9.48E + 02/6.29E + 02	9.43E + 02/5.58E + 01	1.10E + 03/5.84E + 01	1.18E + 03/8.93E + 03	9.67E + 02/2.32E + 03	1.17E + 03/9.88E + 01
All/NDS/DS	7/2/5.	7/5/2	7/5/2	7/1/6	7/4/3	7/0/7

**Table 4** Statistical results (best fitness and cost time) obtained by MGP-SLPSO, SAHO, SACOSO, VSMPSO(GP) and VSMPSO(RBF) on 50D F1–F7 with 550 FEs

Fuction	VSMPSO(GP)	VSMPSO(RBF)	MGP-SLPSO	SAHO	SACOSO
F1: Ellipsoid	9.80E+00/7.98E+03	1.19E+01/1.72E+02	8.53E - 05/1.97E+01	1.22E - 01/3.39E+03	4.93E+01/1.18E+03
F2: Rosenbrock	2.50E+02/1.19E+04	1.88E+02/1.61E+02	1.83E+02/9.86E+01	4.85E+01/1.25E+04	2.49E+02/1.15E+03
F3: Ackley	6.90E+00/1.55E+04	8.68E+00/1.81E+02	1.43E+01/9.01E+01	5.39E - 02/4.86E+03	1.30E+01/2.05E+02
F4: Griewank	2.63E+00/7.70E+03	8.34E - 01/1.80E+02	1.80E - 03/2.11E+01	6.21E - 01/3.72E+03	5.54E+00/1.36E+03
F5: Rastrigin	4.18E+02/1.50E+04	4.04E+02/1.74E+02	2.95E+02/2.17E+02	1.52E+02/6.27E+03	4.24E+02/2.54E+02
F6 (F10 in CEC05)	9.08E+01/8.00E+03	1.05E+02/1.76E+02	1.18E+02/6.42E+02	8.19E+01/7.69E+03	2.14E+02/1.10E+03
F7 (F19 in CEC05)	1.06E+03/1.37E+04	9.81E+02/1.77E+02	1.10E+03/2.16E+03	9.73E+02/6.33E+03	1.08E+03/1.34E+03
All/NDS/DS	7/0/7	7/4/3	7/5/2	7/5/2	7/0/7

shortest running time. In Table 7, MGP-SLPSO achieved 7 NDSs owing to the shortest running time. VSMPSO achieved 5 NDSs same as SACOSO, and SAHO achieved 4 NDSs. VSMPSO did not achieve NDS on F1 and F2, but the DS achieved by VSMPSO is closest to the NDS in Fig. 7. The solutions achieved by VSMPSO, MGP-SLPSO and SAHO are very close to the abscissa value (i.e., the minimum fitness value) in Fig. 7. In Fig. 6, the performance of VSMPSO is stable on different functions, and the convergence curves achieved good results on F5, F6, and F7.

### Numerical results on higher dimensional problems

Next, we compared the algorithm performance on higher-dimensional problems. The convergence profiles of VSMPSO, SACOSO, MGP-SLPSO and SAHO on F1–F7 with 100D and 200D are shown in Figs. 8 and 10. VSMPSO is compared with MGP-SLPSO, CAL-SAPSO, SAHO and SACOSO on 100D. In Table 8, MGP-SLPSO obtains the best mean value on F1 and F4. SAHO obtains the best mean value on F2, F3, F5 and F7. VSMPSO performs the best only on F6. However, on F7, the results of VSMPSO are not significantly different from that of SAHO, and the standard deviation is smaller than SAHO. In Table 9, VSMPSO obtains 5 NDSs, which is the highest NDSs, SAHO achieves 4 NDSs, MGP-SLPSO achieves 3 NDSs, and SACOSO achieves 1 NDS. On F1 and F4, VSMPSO did not achieve NDS, but the DS achieved by VSMPSO is closest to the NDS achieved by MGP-SLPSO in Fig. 9.

Because it takes more than a week for SAHO to obtain the optimization results of one function on 200D problems, it is too expensive to compare VSMPSO with SAHO on 200D problems. So VSMPSO is compared with MGP-SLPSO and SACOSO on 200D problems. In Table 10, VSMPSO obtains the best mean value on F2, F3, and F5, and has a similar performance with SACOSO on F7. SACOSO obtains the best mean values F6 and F7. MGP-SLPSO obtains the best mean value on F1 and F4, and VSMPSO obtains the best mean value second only to MGP-SLPSO on F1 and F4. The running time is no order of magnitude difference between VSMPSO and the MGP-SLPSO, and even the time spent by VSMPSO on F5 is less than MGP-SLPSO. Thus, VSMPSO can achieve a better balance between the optimization effect and the time-consuming and obtains better optimization results on higher dimensional problems (Table 11 and Fig. 11).

### Effects of variable model management strategy

To test and verify the effectiveness of sample selection strategy in VSMPSO, four different sample selection strategies are compared in empirical studies. Therefore, we design a framework of surrogate model-based PSO (SMPSO), in which the RBF model is used as a single surrogate model and

SLPSO as the optimizer. The detailed explanation of SMPSO is located in Algorithm 3. The difference between SMPSO and VSMPSO is the model management strategy. SMPSO only searches for the most promising solution from the current population to evaluate, but VSMPSO takes advantage of the proposed variable model management strategy to search for the most promising solutions from the current population and the current RBF model. SMPSO-RS, SMPSO-AS, SMPSO-FS1 and SMPSO-FS2 are all based on the SMPSO framework; their only difference is in step 7 of the sample selection strategy. The details of different sample selection strategies are as follows:

1. SMPSO-RS: The sample selection strategy is the same as in VSMPSO; simple random sampling (RS) is used for selecting random  $\lambda$  samples in *DB*.
2. ASMP-SO-AS: All samples (AS) in *DB* are chosen for training model, the same as in [32].
3. SMPSO-FS1: Fixed selection of the newest  $\lambda$  samples in *DB*, the same as in [9].
4. SMPSO-FS2: Fixed selection of the top  $\lambda$  samples in *DB*, same as CAL-SAPSO used for training local model.

---

#### Algorithm 4: The pseudocode of SMPSO

---

**Output** : the global best solution with its real fitness

```

1 begin
2    $t = 1$ ;
3   Use LHS to initialize the population and evaluate the swarm
   by the real computationally expensive function;
4   Save all exact evaluations in the Date base DB;
5   while computational budget is not reached the upper limit do
6     Update the particles according to Eqs. (3) and (5);
7     Select  $\lambda$  samples from DB;
8     Train RBF model by the selected samples;
9     Utilize the RBF model to estimate the fitness value of the
   population in generation  $t$ ;
10    Evaluate PGbest by real computationally expensive
   function;
11    Put PGbest with its real fitness into DB;
12    Update the global best solution;
13     $t = t + 1$ ;
14  end while
15 end

```

---

SMPSO-FS1 and SMPSO-FS2 chose fixed sampling (FS) as the sample selection strategy, and the fixed sample number was  $\lambda$ , the same as the sample number in VSMPSO and SMPSO-RS. As noted in “Proposed VSMPSO algorithm” section, the crucial element affecting VSMPSO is the variable model management strategy. The variable model management strategy has two steps: Step 1, a simple random sampling (RS) strategy is used for selecting random  $\lambda$  samples in *DB* to construct the RBF model in each generation.

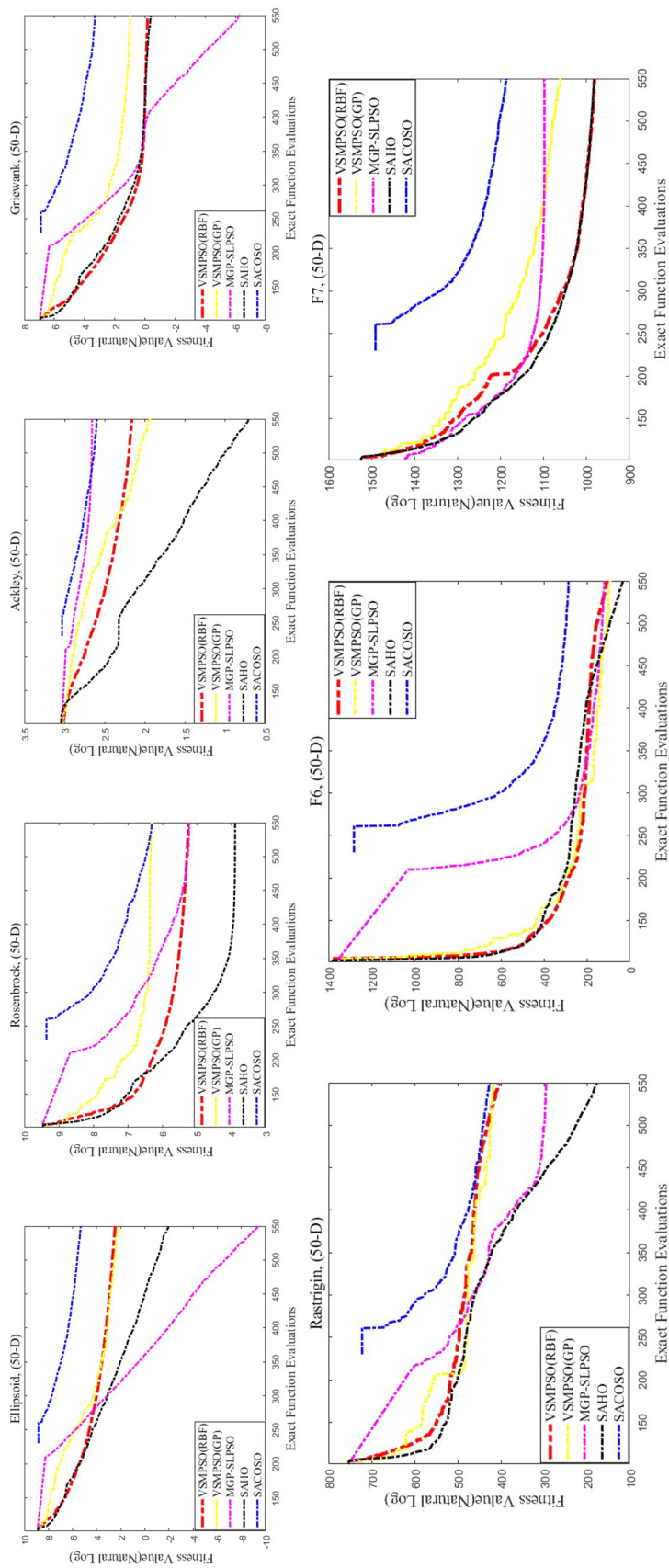


Fig. 4 Convergence profiles of MGP-SLPSO, CAL-SAPSO, SAHO, SACOSO, VSMPSO(RBF) and VSMPSO(GP) on 50D F1–F7 with 550 FEs



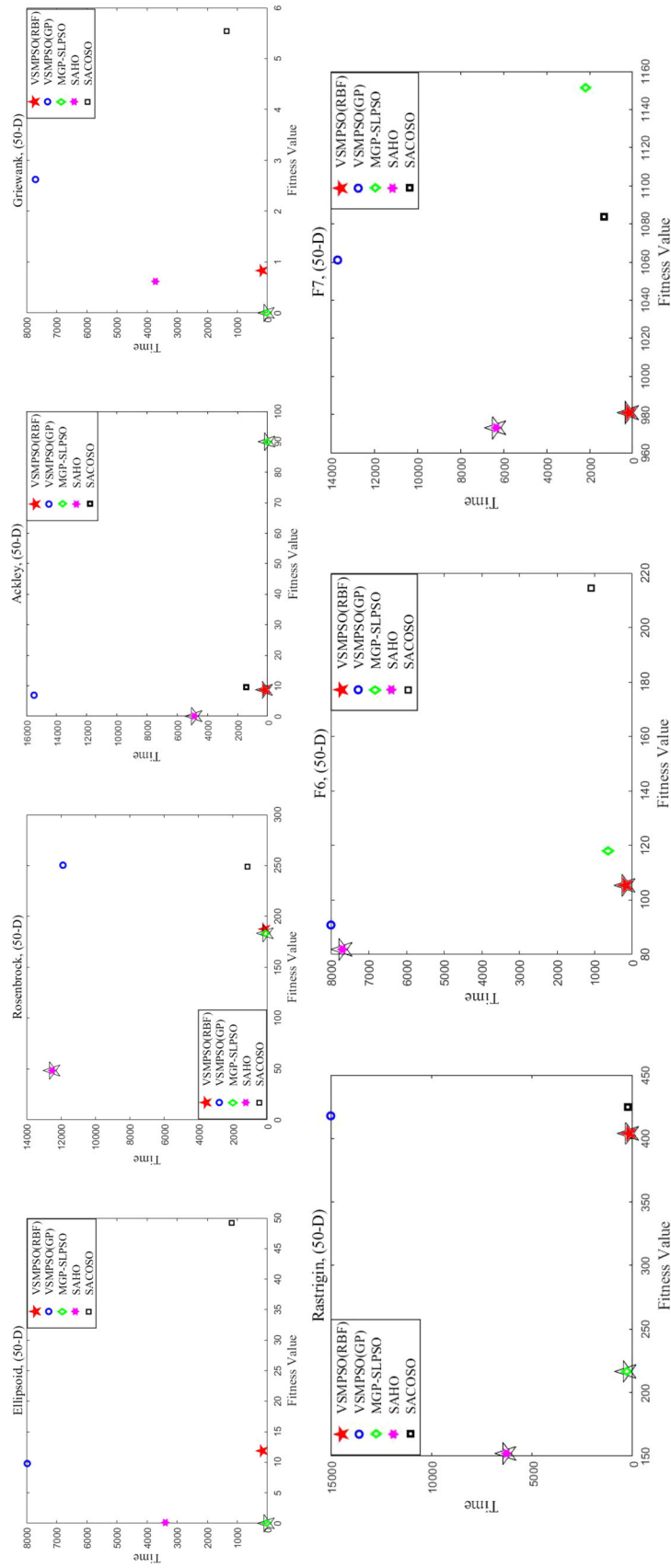


Fig. 5 Distribution of solutions of MGP-SLPSO, CAL-SAPSO, SAHO, SACOSO and VSMPSO on 50D F1–F7 with 550 FEs

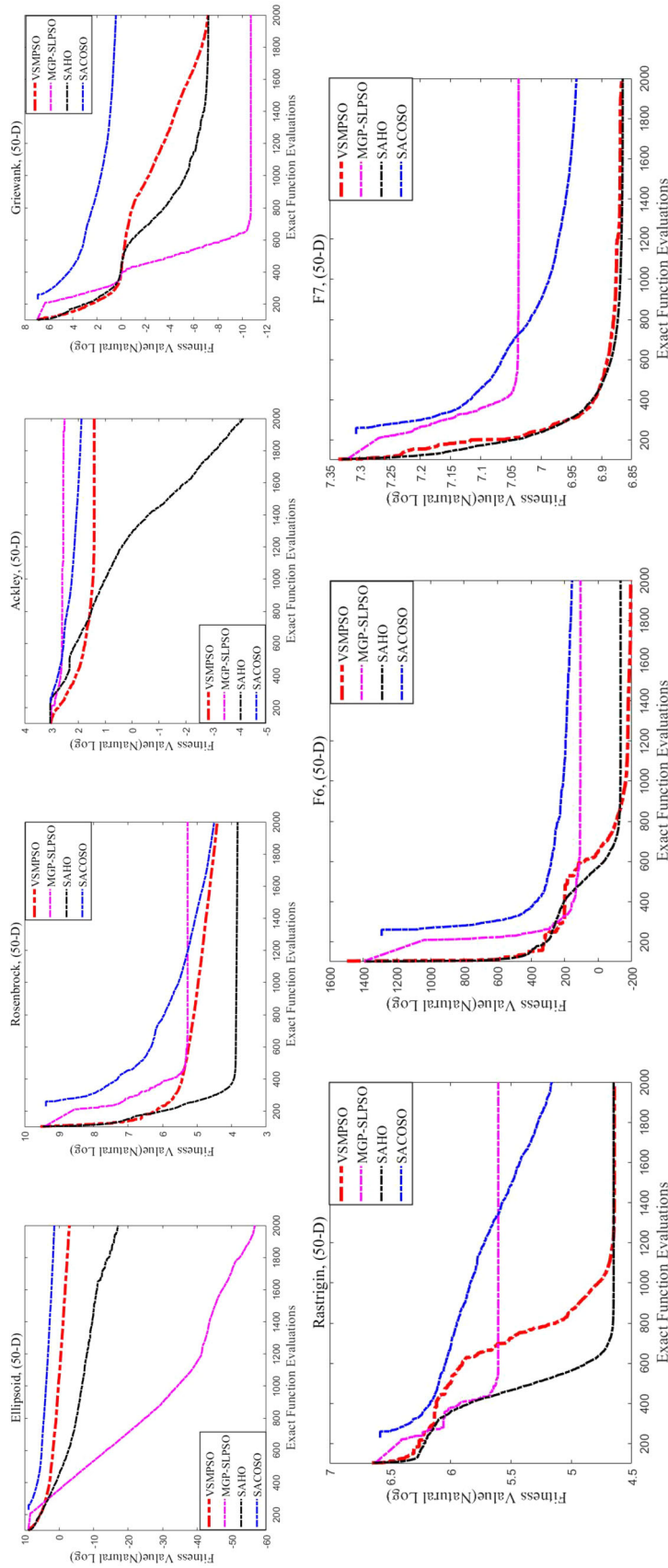


Fig. 6 Convergence profiles of MGP-SLPSO, SAHO, SACOSO and VSMPSO on 50D F1–F7 with 2000 FEs

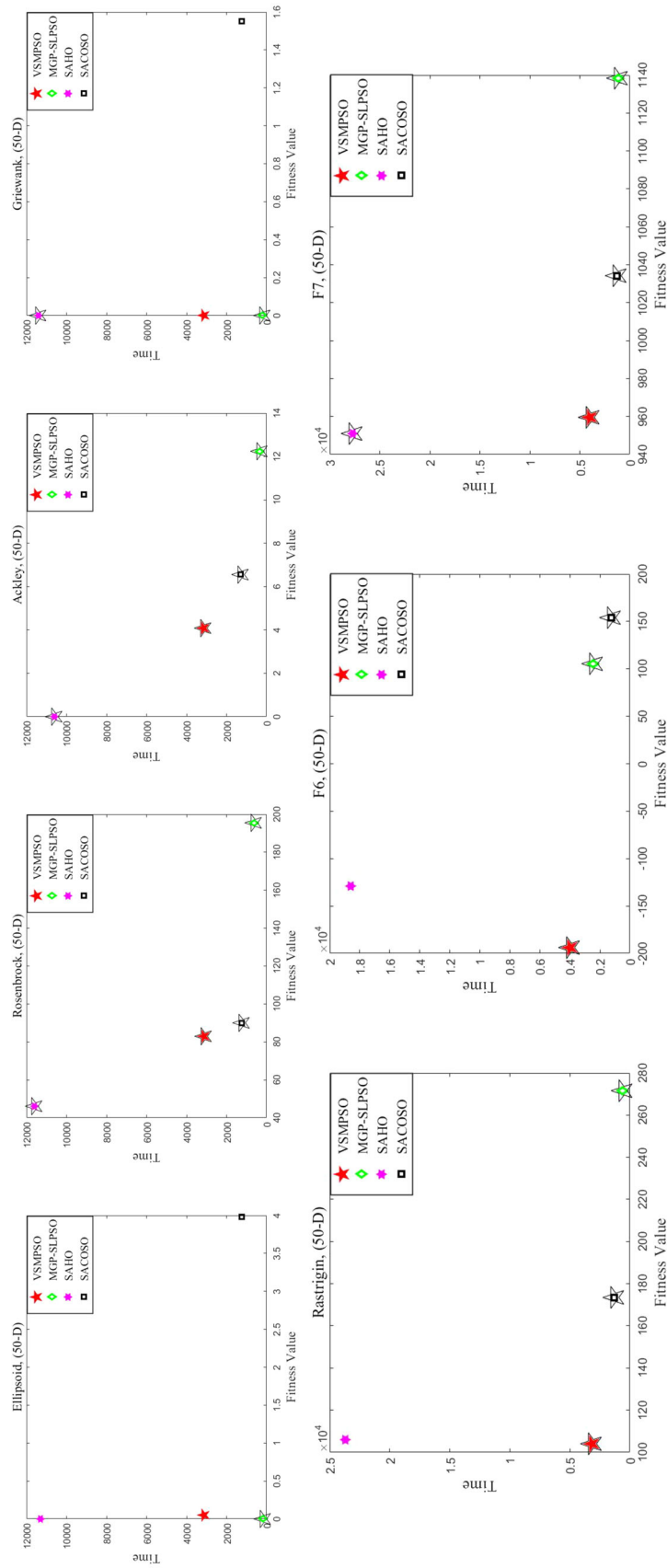


Fig. 7 Distribution of solutions of MGP-SLPSO, SAHO, SACOSO and VSMPSO on 50D F1–F7 with 2000 FEs

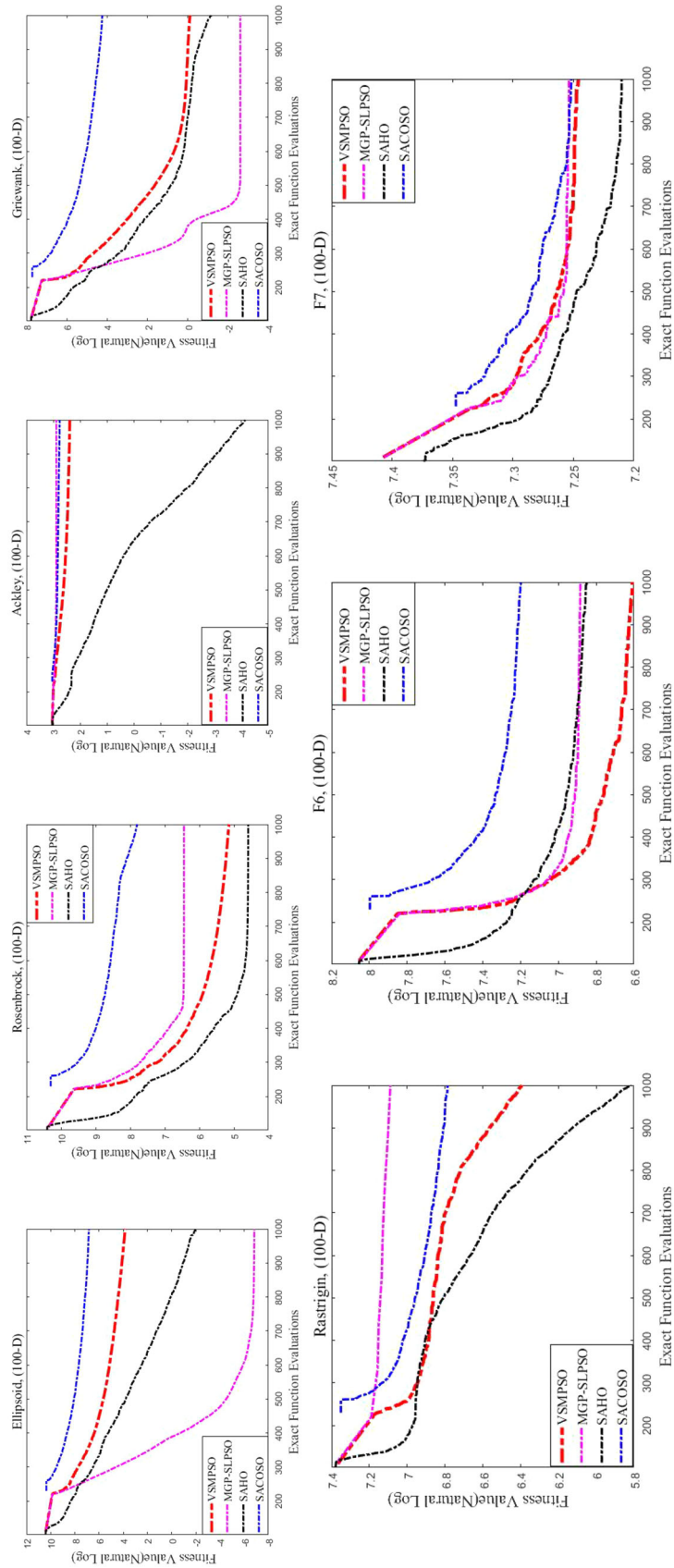


Fig. 8 Convergence profiles of MGP-SLPSO, SAHO, SACOSO and VSMPSO on 100D F1–F7 with 1000 FEs

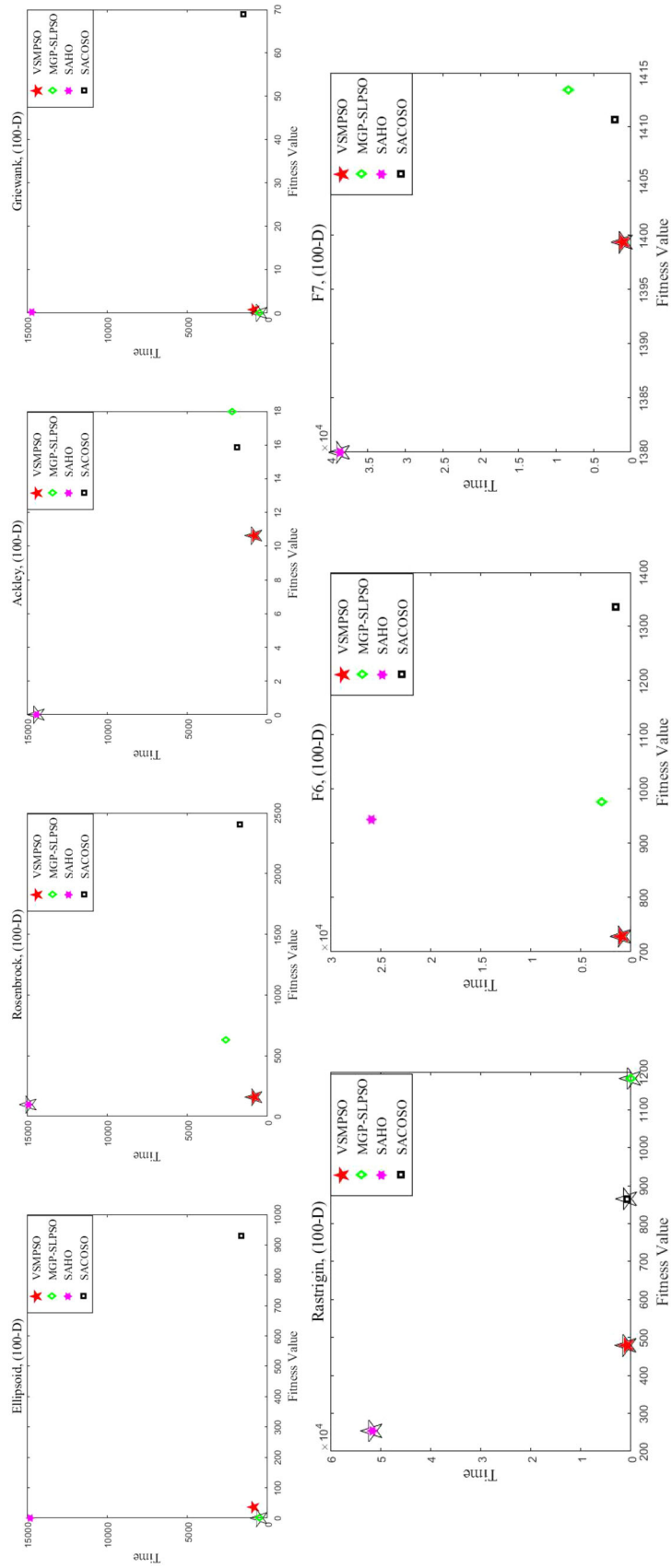


Fig. 9 Distribution of solutions of MGP-SLPSO, SAHO, SACOSO and VSMPSO on 100D F1–F7 with 1000 FEs



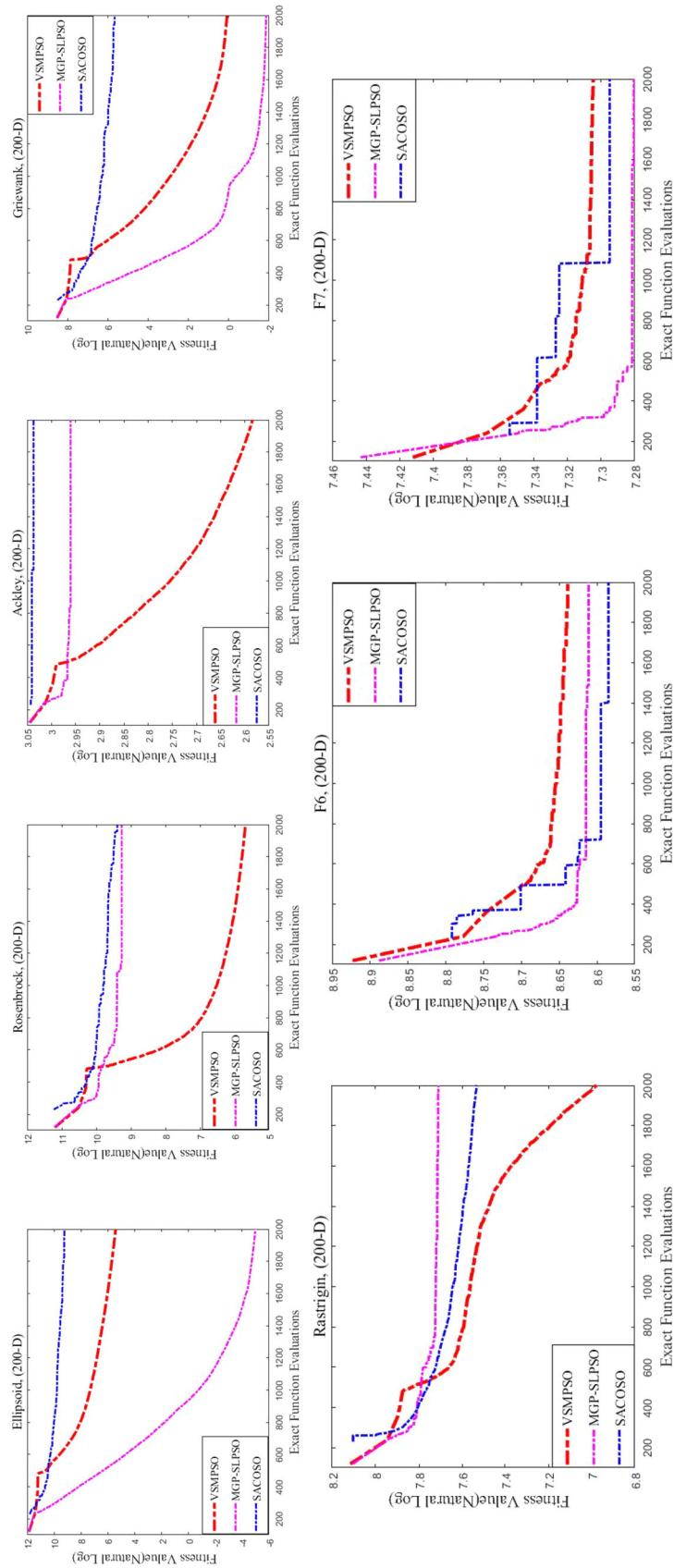


Fig. 10 Convergence profiles of MGP-SLPSO, SACOSO and VSMPSO on 200-D F1–F7 with 2000 FEs

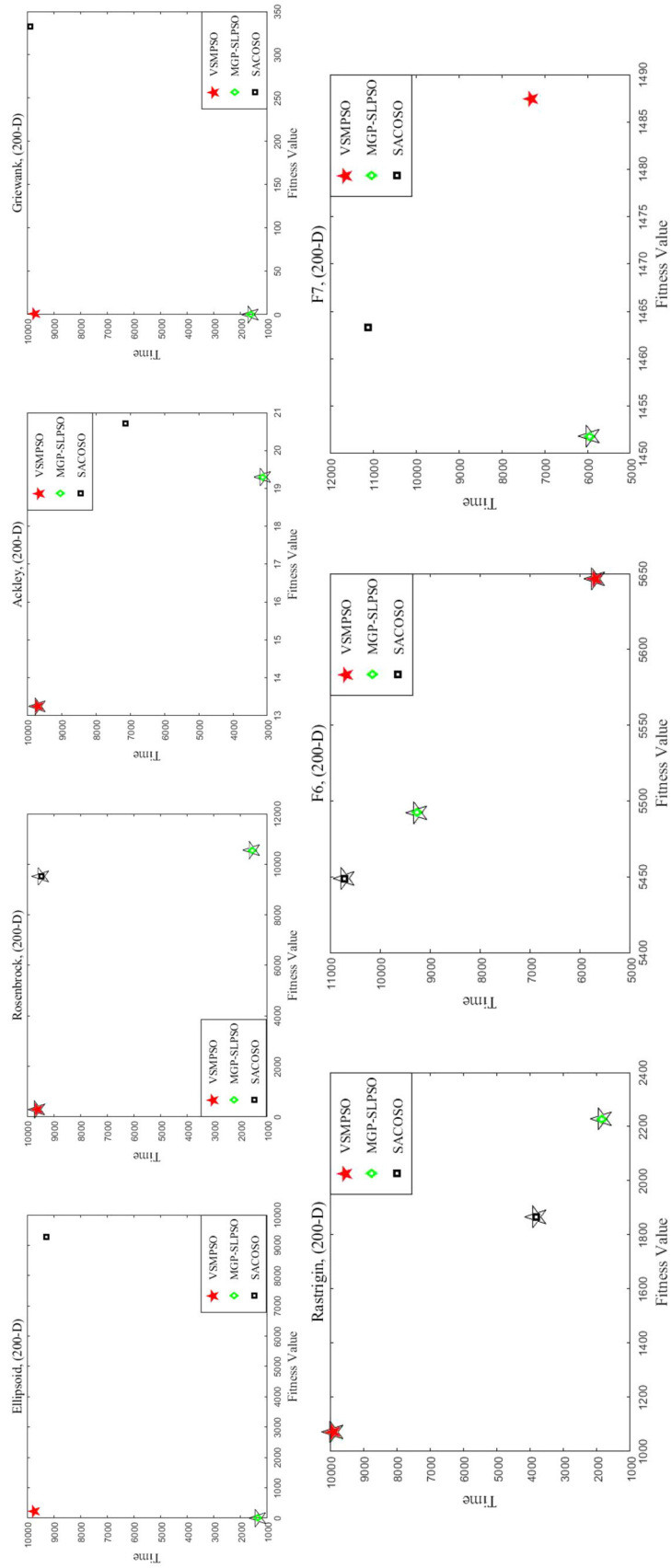


Fig. 11 Distribution of solutions of MGPLPSO, SACOSO and VSMPSO on 200-D F1–F7 with 2000 FEs

**Table 5** Statistical results of MGP-SLPSO, SAHO, SACOSO, VSMP SO(GP) and VSMP SO(RBF) on 50D F1–F7 with 550 FEs

Function	Algorithm	Best	Worst	Mean (Friedman test)	SD
F1: Ellipsoid	VSMP SO(GP)	7.09E+00	1.10E+01	9.80E+00	1.40E+00
	VSMP SO(RBF)	3.79E+00	2.53E+01	1.19E+01	6.13E+00
	MGP-SLPSO	<b>9.31E−06</b>	<b>1.80E−03</b>	<b>8.53E−05(+)</b>	<b>3.23E−04</b>
	SAHO	1.38E−02	6.59E−01	1.22E−01	1.51E−01
	SACOSO	2.38E+01	8.38E+01	4.93E+01	1.60E+01
F2: Rosenbrock	VSMP SO(GP)	4.07E+02	7.36E+02	2.50E+02	1.33E+02
	VSMP SO(RBF)	1.14E+02	2.75E+02	1.88E+02	3.04E+01
	MGP-SLPSO	1.37E+02	2.48E+02	1.83E+02	2.79E+01
	SAHO	<b>4.82E+01</b>	<b>4.87E+01</b>	<b>4.85E+01(+)</b>	<b>1.22E−01</b>
	SACOSO	1.54E+02	3.64E+02	2.49E+02	5.43E+01
F3: Ackley	VSMP SO(GP)	6.52E+00	7.38E+00	6.90E+00	4.21E−01
	VSMP SO(RBF)	4.09E+00	2.00E+01	8.68E+00	6.36E+00
	MGP-SLPSO	1.15E+01	1.57E+01	1.43E+01	1.31E+00
	SAHO	<b>2.38E−02</b>	<b>1.64E−01</b>	<b>5.39E−02(+)</b>	<b>2.97E−02</b>
	SACOSO	1.06E+01	1.46E+01	1.30E+01	1.00E+00
F4: Griewank	VSMP SO(GP)	2.37E+00	3.15E+00	2.63E+00	3.09E−01
	VSMP SO(RBF)	6.42E−01	9.84E−01	8.34E−01	7.38E−02
	MGP-SLPSO	<b>9.51E−04</b>	<b>4.20E−03</b>	<b>1.80E−03(+)</b>	<b>6.79E−04</b>
	SAHO	2.80E−01	9.02E−01	6.21E−01	1.37E−01
	SACOSO	3.69E+00	7.61E+00	5.54E+00	1.04E+00
F5: Rastrigin	VSMP SO(GP)	4.14E+02	4.34E+02	4.18E+02	6.86E+00
	VSMP SO(RBF)	2.50E+02	4.81E+02	4.04E+02	5.97E+01
	MGP-SLPSO	2.05E+02	4.29E+02	2.95E+02	<b>5.78E+01</b>
	SAHO	<b>5.31E+01</b>	<b>4.05E+02</b>	<b>1.52E+02(+)</b>	8.04E+01
	SACOSO	3.58E+02	4.74E+02	4.24E+02	2.99E+01
F6 (F10 in CEC05)	VSMP SO(GP)	7.80E+01	<b>1.21E+02</b>	9.08E+01	<b>1.21E+01</b>
	VSMP SO(RBF)	− 1.15E+01	1.87E+02	1.05E+02	5.72E+01
	MGP-SLPSO	3.35E+01	1.75E+02	1.18E+02	3.74E+01
	SAHO	− <b>1.19E+02</b>	3.51E+02	<b>8.19E+01(+)</b>	1.13E+02
	SACOSO	1.48E+02	2.95E+02	2.14E+02	3.33E+01
F7 (F19 in CEC05)	VSMP SO(GP)	1.02E+03	1.10E+03	1.06E+03	3.46E+01
	VSMP SO(RBF)	<b>9.49E+02</b>	1.04E+03	9.81E+02 (≈)	2.49E+01
	MGP-SLPSO	1.02E+03	1.17E+03	1.10E+03	4.15E+01
	SAHO	9.53E+02	<b>1.03E+03</b>	<b>9.73E+02(+)</b>	<b>1.91E+01</b>
	SACOSO	1.00E+03	1.16E+03	1.08E+03	3.66E+01

Step 2, the current RBF model information is deeply mined by finding the minimum of the surrogate model. Two potential optimum points ( $PG_{best}$  and  $MG_{best}$ ) are considered in the variable model management strategy.  $PG_{worst}$ , the individual with the worst predicted fitness value in the current population, is then replaced by  $MG_{best}$ . The following experiment results further demonstrate that this is an effective method to improve the diversity of the model and avoid local optimum without using multiple models.

The contribution of the simple random sampling strategy in VSMP SO is inferred by comparing the results of

SMP SO-RS with SMP SO-AS, SMP SO-FS1, and SMP SO-FS2. From Table 12, on 30D functions, SMP SO-RS achieved better optimization results on F3 and F7 than SMP SO-AS, SMP SO-FS1, and SMP SO-FS2, and the optimization results of SMP SO-RS on F2, F4, F5, and F6 were second only to SMP SO-AS. From Fig. 12, the trend of convergence curves is consistent, except that the F1 and F3 convergence curves are slightly dispersed. From Table 12, on 50D functions, SMP SO-RS achieved better optimization results on F2, F4, F6, and F7 than SMP SO-AS, SMP SO-FS1, and SMP SO-FS2, and the optimization results of SMP SO-RS on F1, F3,

**Table 6** Statistical results of MGP-SLPSO, SAHO, SACOSO and VSMPSO on 50D F1–F7 with 2000 FEs

Function	Algorithm	Best	Worst	Mean (Friedman test)	SD
F1: Ellipsoid	VSMPSO	4.01E−02	7.60E−02	5.05E−02	1.05E−02
	MGP-SLPSO	<b>4.72E−26</b>	<b>1.90E−24</b>	<b>2.47E−25(+)</b>	<b>5.82E−25</b>
	SAHO	6.75E−16	4.41E−07	3.49E−10	9.94E−08
	SACOSO	1.36E+00	6.90E+00	3.98E+00	1.45E+00
F2: Rosenbrock	VSMPSO	5.92E+01	1.18E+02	8.30E+01	2.47E+01
	MGP-SLPSO	1.78E+02	2.19E+02	1.95E+02	1.40E+01
	SAHO	<b>4.56E+01</b>	<b>4.63E+01</b>	<b>4.61E+01(+)</b>	<b>1.42E−01</b>
	SACOSO	5.41E+01	1.85E+02	9.01E+01	3.24E+01
F3: Ackley	VSMPSO	3.18E+00	5.38E+00	4.09E+00	7.89E−01
	MGP-SLPSO	9.44E+02	9.86E+02	9.60E+02	1.18E+01
	SAHO	<b>1.29E−12</b>	<b>2.14E−11</b>	<b>4.49E−12(+)</b>	<b>4.72E−12</b>
	SACOSO	1.06E+01	1.46E+01	1.30E+01	1.00E+00
F4: Griewank	VSMPSO	2.71E−04	1.80E−03	7.50E−04	7.00E−04
	MGP-SLPSO	5.57E−09	<b>8.65E−05</b>	2.19E−05	<b>3.61E−05</b>
	SAHO	<b>1.30E−11</b>	1.24E−02	<b>3.73E−10(+)</b>	2.84E−03
	SACOSO	3.69E+00	7.61E+00	5.54E+00	1.04E+00
F5: Rastrigin	VSMPSO	<b>8.76E+01</b>	<b>1.45E+02</b>	<b>1.04E+02(+)</b>	<b>1.60E+01</b>
	MGP-SLPSO	1.91E+02	4.15E+02	2.72E+02	7.33E+01
	SAHO	4.88E+01	1.63E+02	1.06E+02	2.80E+01
	SACOSO	9.11E+01	2.72E+02	1.73E+02	3.81E+01
F6 (F10 in CEC05)	VSMPSO	<b>− 2.28E+02</b>	<b>− 1.79E+02</b>	<b>− 1.94E+02(+)</b>	<b>1.39E+01</b>
	MGP-SLPSO	5.80E+01	1.37E+02	1.05E+02	2.85E+01
	SAHO	<b>− 2.25E+02</b>	<b>− 1.84E+01</b>	<b>− 1.29E+02</b>	4.95E+01
	SACOSO	<b>− 2.58E+01</b>	2.51E+02	1.54E+02	4.97E+01
F7 (F19 in CEC05)	VSMPSO	9.44E+02	<b>9.86E+02</b>	9.60E+02(≈)	<b>1.18E+01</b>
	MGP-SLPSO	1.08E+03	1.19E+03	1.14E+03	3.38E+01
	SAHO	<b>9.39E+02</b>	1.03E+03	<b>9.51E+02(+)</b>	2.12E+01
	SACOSO	9.81E+02	1.12E+03	1.03E+03	3.22E+01

**Table 7** Statistical results (best fitness and cost time) obtained by MGP-SLPSO, SAHO, SACOSO and VSMPSO on 50D F1–F7 With 2000 FEs

Mean/time	VSMPSO	MGP-SLPSO	SAHO	SACOSO
F1: Ellipsoid	5.05E−02/3.14E+03	2.47E−25/1.82E+02	3.49E−10/1.13E+04	3.98E+00/1.25E+03
F2: Rosenbrock	8.30E+01/3.14E+03	1.95E+02/6.34E+02	4.61E+01/1.16E+04	9.01E+01/1.24E+03
F3: Ackley	4.09E+00/3.15E+03	9.60E+02/3.44E+02	4.49E−12/1.06E+04	1.30E+01/2.05E+02
F4: Griewank	7.50E−04/3.14E+03	2.19E−05/2.03E+02	3.73E−10/1.14E+04	5.54E+00/1.36E+03
F5: Rastrigin	1.04E+02/3.15E+03	1.04E+02/6.02E+02	1.06E+02/2.37E+04	1.73E+02/1.29E+03
F6 (F10 in CEC05)	<b>− 1.94E+02/3.98E+03</b>	1.05E+02/2.45E+03	1.29E+02/1.86E+04	1.54E+02/1.25E+03
F7 (F19 in CEC05)	9.60E+02/4.01E+03	1.14E+03/1.19E+03	9.51E+02/2.77E+04	1.03E+03/1.31E+03
All/NDS/DS	7/5/2	<b>7/7/0</b>	7/4/3	7/5/2

**Table 8** Statistical results of MGP-SLPSO, SAHO, SACOSO and VSMPSO on 100D F1–F7 with 1000 FEs

Function	Algorithm	Best	Worst	Mean (Friedman test)	SD
F1: Ellipsoid	VSMPSO	2.17E+01	6.20E+01	3.60E+01	9.31E+00
	MGP-SLPSO	<b>4.37E−17</b>	<b>3.17E−02</b>	<b>1.10E−03(+)</b>	<b>5.80E−03</b>
	SAHO	2.80E−02	1.99E+00	5.02E−02	3.55E−01
	SACOSO	5.94E+02	1.39E+03	9.29E+02	2.36E+02
F2: Rosenbrock	VSMPSO	1.53E+02	2.40E+02	1.88E+02	2.07E+01
	MGP-SLPSO	4.58E+02	8.59E+02	6.28E+02	1.05E+02
	SAHO	<b>9.78E+01</b>	<b>9.82E+01</b>	<b>9.80E+01(+)</b>	<b>7.37E−02</b>
	SACOSO	1.46E+03	4.64E+03	2.41E+03	7.99E+02
F3: Ackley	VSMPSO	8.24E+00	1.25E+01	1.06E+01	8.99E−01
	MGP-SLPSO	1.71E+01	1.85E+01	1.80E+01	3.47E−01
	SAHO	<b>9.60E−03</b>	<b>2.94E−02</b>	<b>1.48E−02(+)</b>	<b>4.84E−03</b>
	SACOSO	1.42E+01	1.72E+01	1.59E+01	7.44E−01
F4: Griewank	VSMPSO	6.58E−01	1.01E+00	8.17E−01	8.24E−02
	MGP-SLPSO	<b>3.10E−02</b>	<b>1.31E−01</b>	<b>7.37E−02(+)</b>	<b>2.51E−02</b>
	SAHO	1.02E−01	6.52E−01	2.51E−01	1.54E−01
	SACOSO	4.14E+01	1.06E+02	6.90E+01	1.50E+01
F5: Rastrigin	VSMPSO	3.09E+02	7.32E+02	4.79E+02	8.71E+01
	MGP-SLPSO	1.09E+03	1.24E+03	1.18E+03	<b>3.52E+01</b>
	SAHO	<b>1.23E+02</b>	<b>3.85E+02</b>	<b>2.54E+02(+)</b>	6.01E+01
	SACOSO	7.84E+02	9.63E+02	8.65E+02	4.87E+01
F6 (F10 in CEC05)	VSMPSO	<b>5.86E+02</b>	<b>8.36E+02</b>	<b>7.28E+02(+)</b>	<b>6.55E+01</b>
	MGP-SLPSO	7.79E+02	1.38E+03	9.77E+02	1.09E+02
	SAHO	7.39E+02	1.17E+03	9.44E+02	9.05E+01
	SACOSO	1.10E+03	1.60E+03	1.34E+03	1.13E+02
F7 (F19 in CEC05)	VSMPSO	1.33E+03	1.46E+03	1.40E+03 (≈)	4.05E+01
	MGP-SLPSO	1.35E+03	1.54E+03	1.41E+03 (≈)	4.12E+01
	SAHO	<b>9.10E+02</b>	<b>1.42E+03</b>	<b>1.38E+03(+)</b>	1.22E+02
	SACOSO	1.35E+03	1.52E+03	1.41E+03 (≈)	<b>3.80E+01</b>

**Table 9** Statistical results (best fitness and cost time) obtained by MGP-SLPSO, SAHO, SACOSO and VSMPSO on 100D F1–F7 with 1000 FEs

Mean/time	VSMPSO	MGP-SLPSO	SAHO	SACOSO
Ellipsoid	3.60E+01/8.71E+02	1.10E−03/5.17E+02	5.02E−02/1.48E+04	1.10E−03/5.17E+02
Rosenbrock	1.88E+02/6.07E+02	6.28E+02/2.62E+03	9.80E+01/1.49E+04	6.28E+02/2.62E+03
Ackley	1.06E+01/8.33E+02	1.80E+01/2.20E+03	1.48E−02/1.44E+04	1.80E+01/2.20E+03
Griewank	8.17E−01/8.49E+02	7.37E−02/5.21E+02	2.51E−01/1.47E+04	7.37E−02/5.21E+02
Rastrigin	4.79E+02/8.80E+02	1.18E+03/4.01E−02	2.54E+02/5.16E+04	8.65E+02/7.76E+02
F6 (F10 in CEC05)	7.28E+02/9.45E+02	9.77E+02/2.93E+03	9.44E+02/2.59E+04	9.77E+02/2.93E+03
F7 (F19 in CEC05)	1.40E+03/1.11E+03	1.41E+03/8.30E+03	1.38E+03/3.87E+04	1.41E+03/8.30E+03
All/NDS/DS	<b>7/5/2</b>	7/3/4	7/4/3	7/1/6



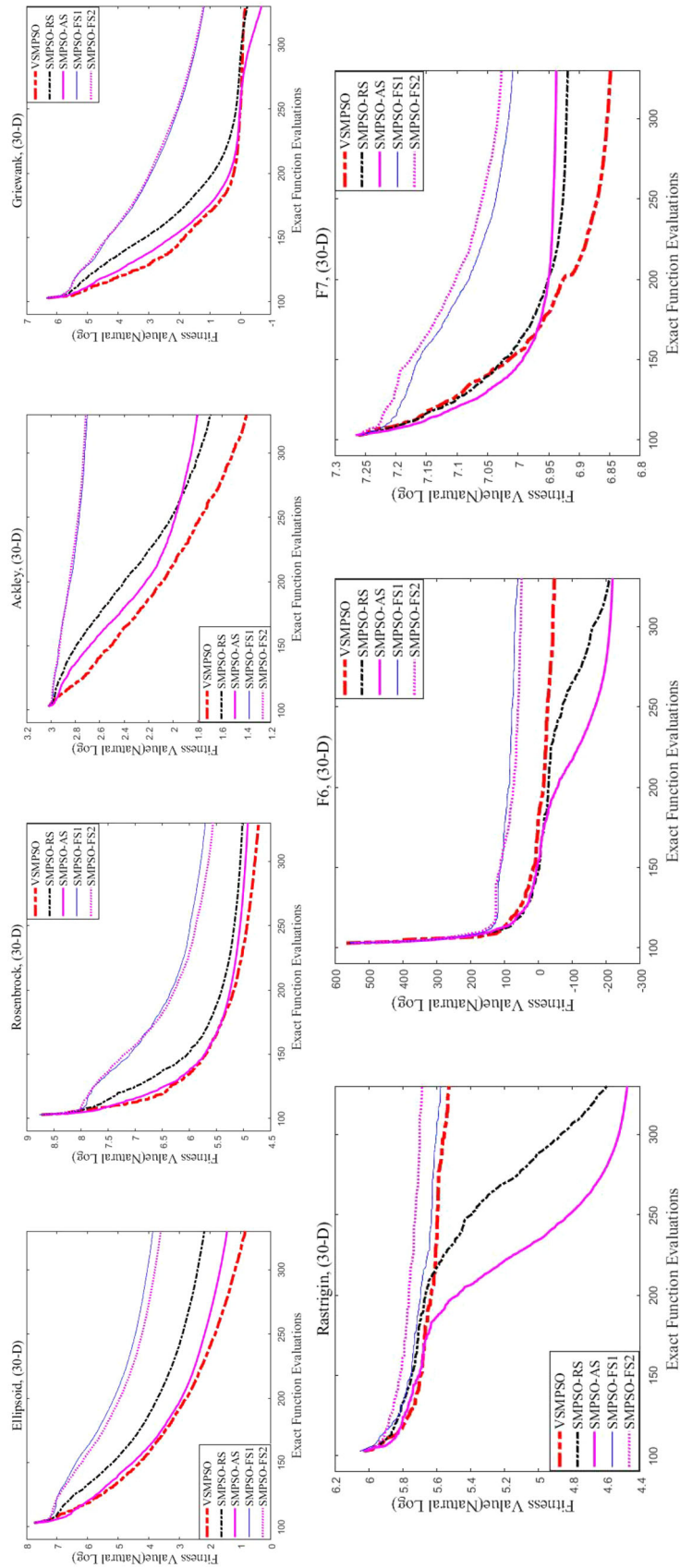


Fig. 12 Convergence profiles of SMPSO-RS, SMPSO-AS, SMPSO-FS1, SMPSO-FS2, VSMPSO on 30-D F1–F7 with 330 FEs

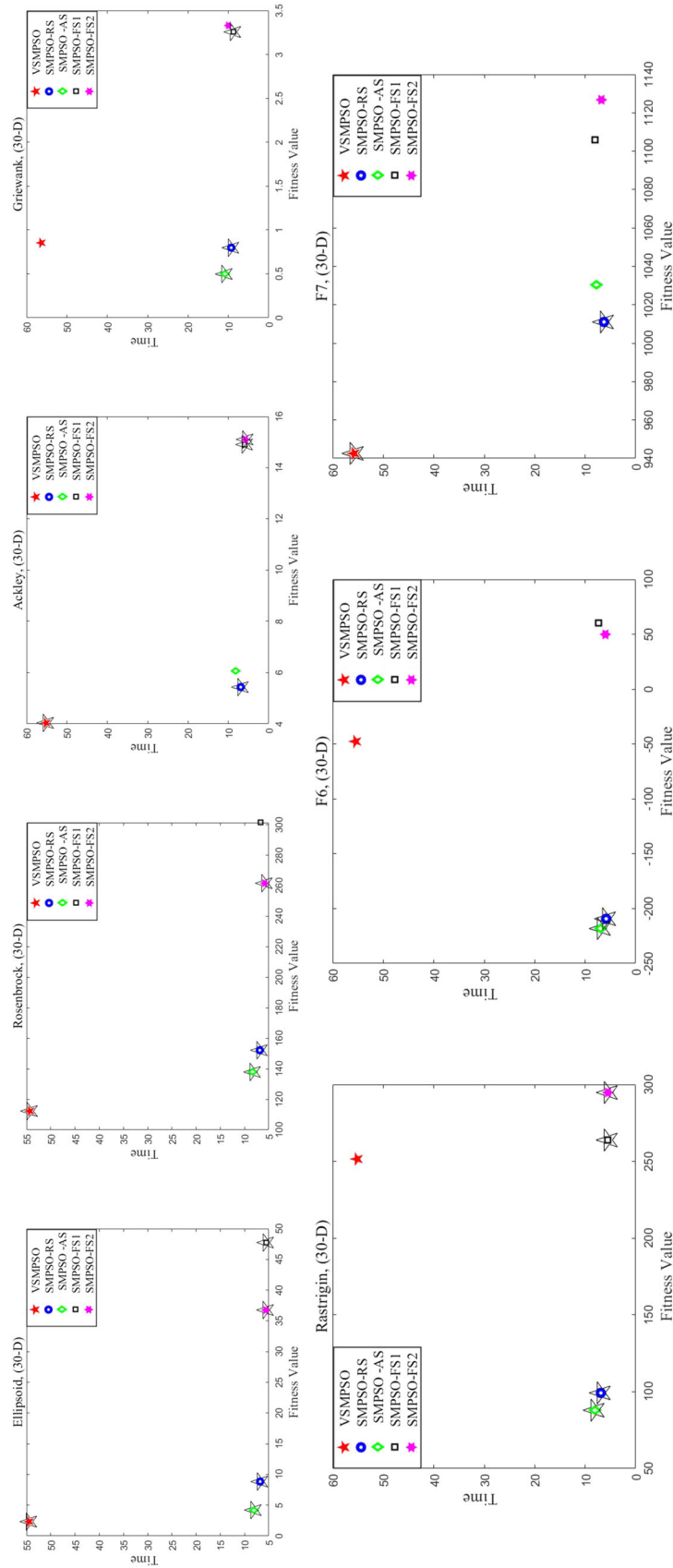


Fig. 13 Distribution of solutions of SMPSO-RS, SMPSO-AS, SMPSO-FS1, SMPSO-FS2 and VSMPSO on 30-D F1–F7 with 330 FEs

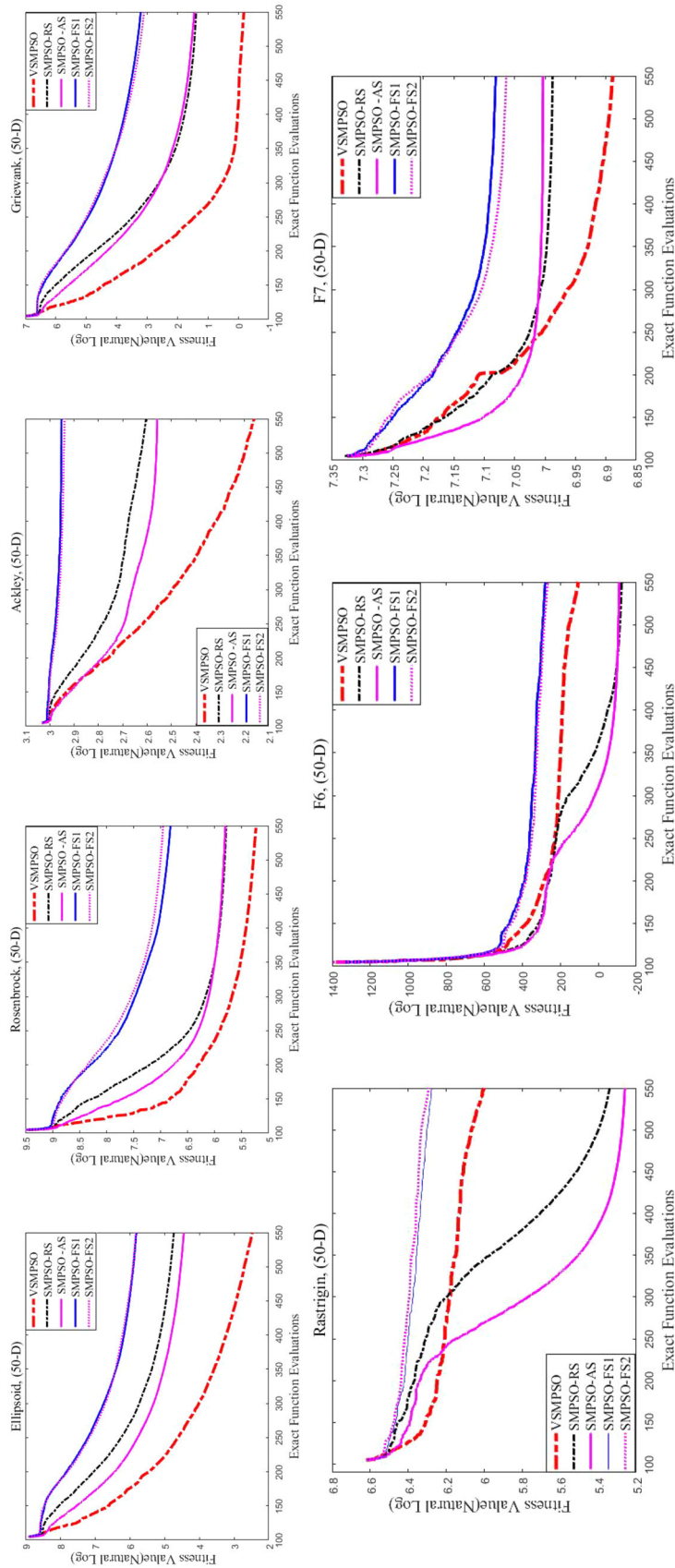


Fig. 14 Convergence profiles of SMPSO-RS, SMPSO-AS, SMPSO-FS1, SMPSO-FS2 and VSMPSO on 50-D F1–F7 with 550 FE

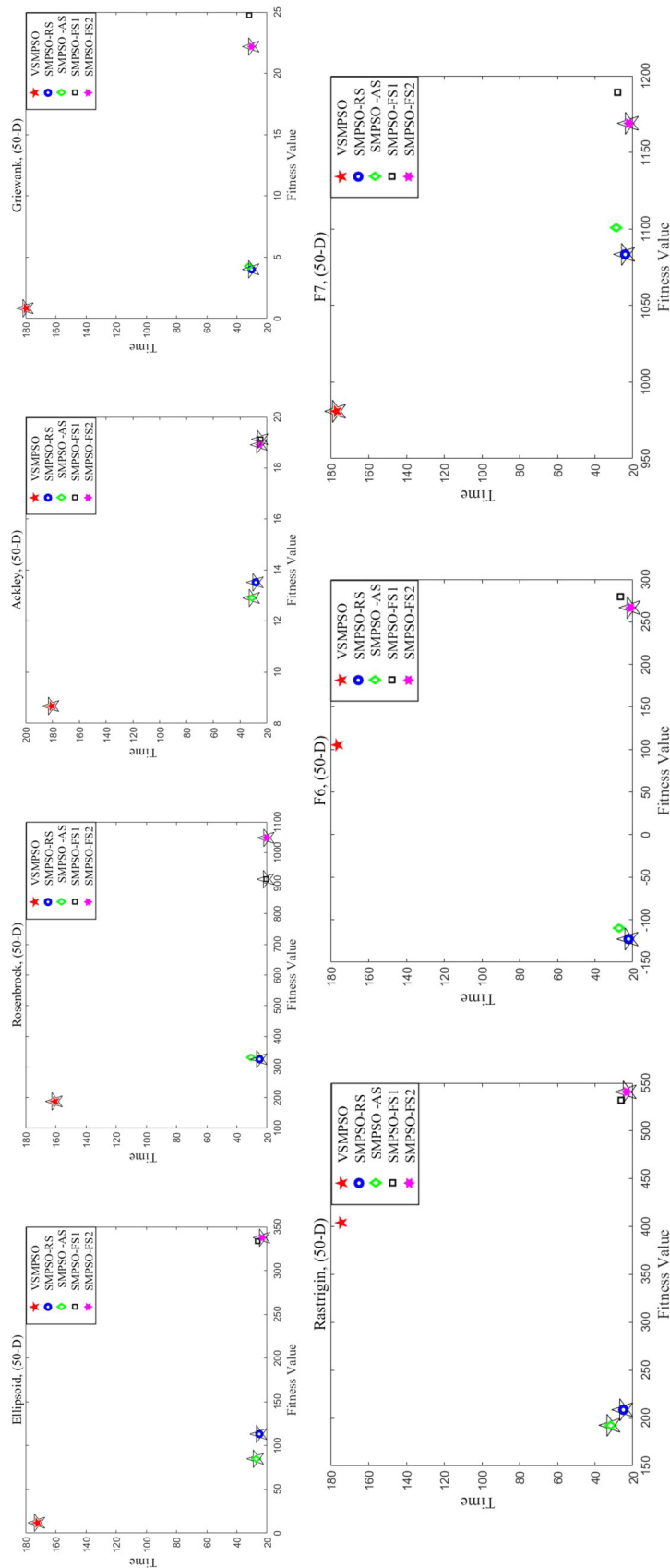


Fig. 15 Distribution of solutions of SMPSO-RS, SMPSO-AS, SMPSO-FS1, SMPSO-FS2 and VSMPSo on 50-D F1–F7 with 550 FEs

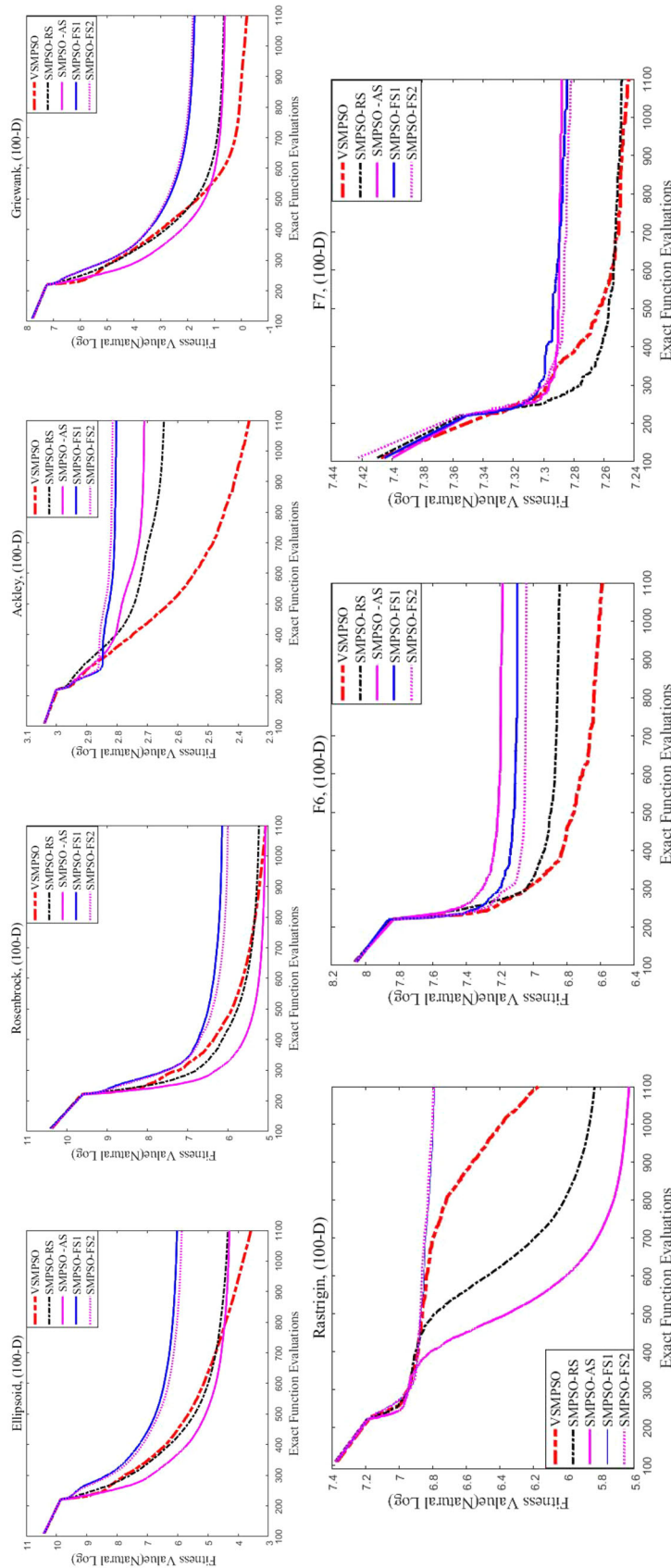


Fig. 16 Convergence profiles of SMPSO-RS, SMPSO-AS, SMPSO-FS1, SMPSO-FS2 and VSMPSO on 100-D F1–F7 with 1100 FEs

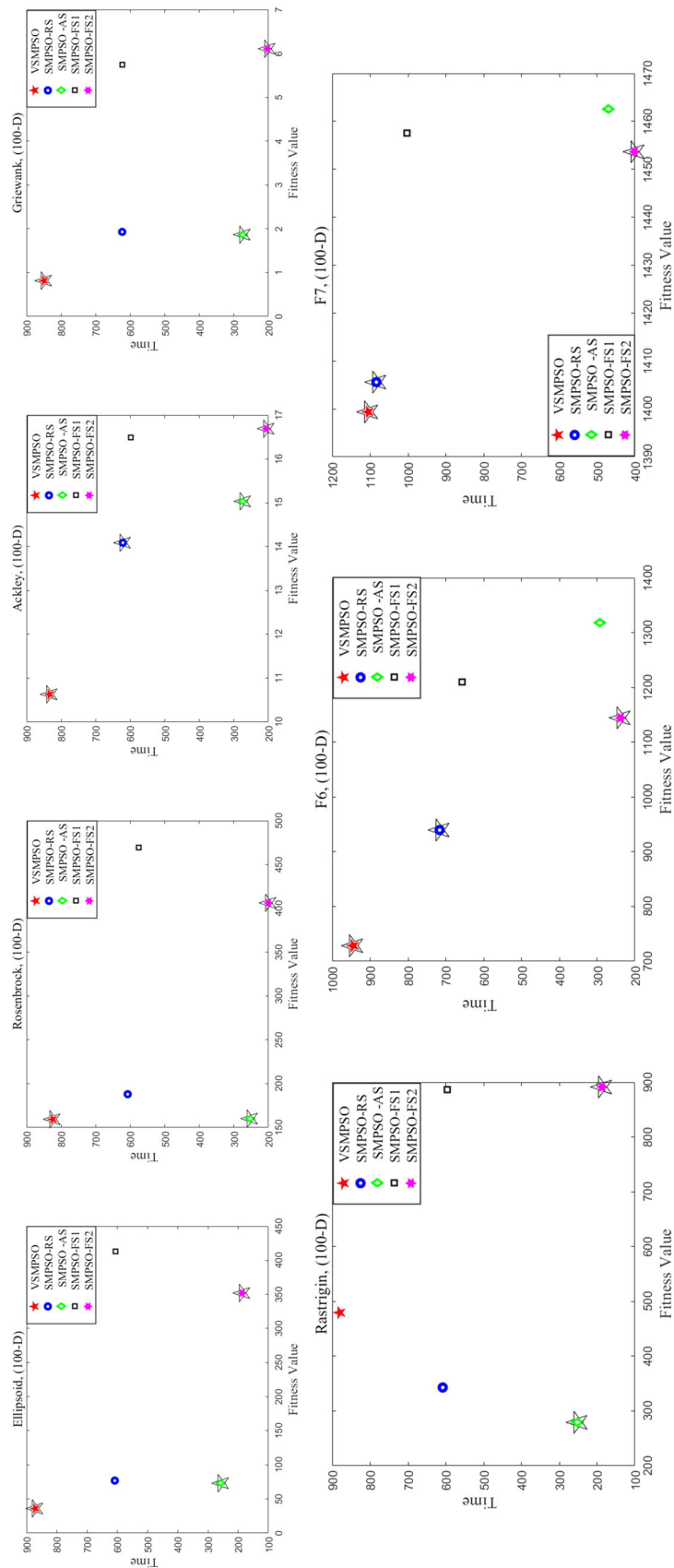


Fig. 17 Distribution of solutions of SMPSO-RS, SMPSO-AS, SMPSO-FS1, SMPSO-FS2 and VSMPSO on 100-D F1–F7 with 1100 FEs



**Table 10** Statistical results of MGP-SLPSO, SACOSO and VSMPSO on 200D F1–F7 with 2000 FEs

Function	Algorithm	Best	Worst	Mean	SD
F1: Ellipsoid	VSMPSO	1.63E+02	2.99E+02	2.25E+02	3.48E+01
	MGP-SLPSO	<b>2.51E−04</b>	<b>1.21E−02</b>	<b>6.80E−03(+)</b>	<b>3.10E−03</b>
	SACOSO	6.07E+03	1.47E+04	9.27E+03	2.12E+03
F2: Rosenbrock	VSMPSO	<b>2.68E+02</b>	<b>3.20E+02</b>	<b>2.94E+02(+)</b>	<b>1.45E+01</b>
	MGP-SLPSO	7.96E+03	1.20E+04	1.06E+04	1.62E+03
	SACOSO	4.22E+03	1.76E+04	9.52E+03	3.06E+03
F3: Ackley	VSMPSO	<b>1.21E+01</b>	<b>1.42E+01</b>	<b>1.33E+01(+)</b>	5.48E−01
	MGP-SLPSO	1.47E+03	1.52E+03	1.49E+03	1.28E+01
	SACOSO	2.05E+01	2.09E+01	2.07E+01	<b>1.01E−01</b>
F4: Griewank	VSMPSO	1.01E+00	1.18E+00	1.09E+00	<b>3.96E−02</b>
	MGP-SLPSO	<b>4.91E−05</b>	<b>1.09E+00</b>	<b>1.57E−01(+)</b>	3.30E−01
	SACOSO	2.19E+02	5.01E+02	3.33E+02	7.54E+01
F5: Rastrigin	VSMPSO	<b>8.64E+02</b>	<b>1.35E+03</b>	<b>1.07E+03(+)</b>	1.35E+02
	MGP-SLPSO	2.15E+03	2.37E+03	2.23E+03	8.41E+01
	SACOSO	1.68E+03	2.12E+03	1.87E+03	<b>7.42E+01</b>
F6 (F10 in CEC05)	VSMPSO	5.10E+03	5.92E+03	5.65E+03	1.98E+02
	MGP-SLPSO	<b>5.06E+03</b>	<b>5.76E+03</b>	5.49E+03	1.93E+02
	SACOSO	5.12E+03	5.65E+03	<b>5.45E+03(+)</b>	<b>1.30E+02</b>
F7 (F19 in CEC05)	VSMPSO	1.47E+03	1.52E+03	1.49E+03	1.28E+01
	MGP-SLPSO	1.44E+03	<b>1.46E+03</b>	<b>1.45E+03(+)</b>	<b>1.17E+01</b>
	SACOSO	<b>1.41E+03</b>	1.49E+03	1.46E+03(≈)	1.67E+01

**Table 11** Statistical results (best fitness and cost time) obtained by MGP-SLPSO, SACOSO and VSMPSO on 200-D F1–F7 with 2000 FEs

Function	VSMPSO	MGP-SLPSO	SACOSO
F1: Ellipsoid	2.25E+02/9.73E+03	6.80E−03/1.34E+03	9.27E+03/9.29E+03
F2: Rosenbrock	2.94E+02/9.62E+03	1.06E+04/1.57E+03	9.52E+03/9.48E+03
F3: Ackley	1.33E+01/9.69E+03	1.49E+03/3.12E+03	2.07E+01/7.14E+03
F4: Griewank	1.09E+00/9.72E+03	1.57E−01/1.62E+03	3.33E+02/9.88E+03
F5: Rastrigin	1.07E+03/9.90E+03	2.23E+03/1.85E+03	1.87E+03/3.82E+03
F6 (F10 in CEC05)	5.65E+03/5.70E+03	5.49E+03/9.25E+03	5.45E+03/1.07E+04
F7 (F19 in CEC05)	1.49E+03/7.31E+03	1.45E+03/5.94E+03	1.46E+03/1.11E+04
All/NDS/DS	7/4/3	7/7/0	7/3/4

and F5 were slightly less than SMPSO-AS but better than SMPSO-FS1 and SMPSO-FS2. In addition, from Fig. 14, the trend of convergence curves for SMPSO-RS are very similar to that of SMPSO-AS on most functions.

From Table 15, SMPSO-RS achieved better optimization results on F3, F6, and F7 than SMPSO-AS, SMPSO-FS1, and SMPSO-FS2, and the optimization results of SMPSO-RS on F1, F2, F4, and F5 were slightly lower than SMPSO-AS but better than SMPSO-FS1 and SMPSO-FS2. Moreover, from Figs. 13, 15 and Table 14, SMPSO-RS achieved non-dominant solutions for both 30D and 50D problems. From Fig. 17, the dominant solution obtained by SMPSO-RS is closest to the non-dominant solution in abscissa value (that is the minimum fitness value) on F1, F2, F4, and F5. Conversely, SMPSO-FS1 and SMPSO-FS2 have similar results

on Tables 12, 13 and 15, and show similar convergence curve on Figs. 12, 14 and 16. Moreover, both SMPSO-FS1 and SMPSO-FS2 have worse results than SMPSO-RS and SMPSO-AS; hence, the performance of fixed sampling methods is inferior to all sampling methods and the simple random sampling method. The performance of simple random sampling (RS) is significantly better than that of fixed sampling (FS) and SMPSO-RS performs better on complex problems. Conversely, SMPSO-AS takes more time than SMPSO-RS on 30D and 50D functions. When all samples are used for modeling in SMPSO-AS, it is time-consuming.

By comparing VSMPSO with SMPSO-RS, SMPSO-AS, SMPSO-FS1 and SMPSO-FS2, we can see the contribution of Step 2. In Step2, VSMPSO takes advantage of the variable model management strategy to search for the most

**Table 12** Statistical results of VSMP SO-RS, VSMP SO-AS, VSMP SO-FS1, VSMP SO-FS2, VSMP SO on 30-D with 330 FEs

Function	Algorithm	Best	Worst	Mean (Friedman test)	SD
F1: Ellipsoid	SMP SO-RS	1.67E+00	3.42E+01	8.84E+00	7.73E+00
	SMP SO-AS	3.54E−01	1.55E+01	4.21E+00	3.53E+00
	SMP SO-FS1	8.51E+00	1.22E+02	4.77E+01	3.16E+01
	SMP SO-FS2	1.14E+01	9.31E+01	3.68E+01	2.17E+01
	VSMP SO	<b>2.67E−01</b>	<b>9.20E+00</b>	<b>2.33E+00(+)</b>	<b>1.81E+00</b>
F2: Rosenbrock	SMP SO-RS	1.03E+02	2.13E+02	1.52E+02	<b>2.80E+01</b>
	SMP SO-AS	8.35E+01	<b>1.92E+02</b>	1.38E+02	3.05E+01
	SMP SO-FS1	1.54E+02	9.40E+02	3.01E+02	1.79E+02
	SMP SO-FS2	1.53E+02	5.69E+02	2.62E+02	9.14E+01
	VSMP SO	<b>5.99E+01</b>	2.03E+02	<b>1.12E+02(+)</b>	3.01E+01
F3: Ackley	SMP SO-RS	<b>3.21E+00</b>	1.52E+01	5.43E+00	2.45E+00
	SMP SO-AS	3.40E+00	1.16E+01	6.07E+00	2.13E+00
	SMP SO-FS1	9.57E+00	1.90E+01	1.49E+01	2.77E+00
	SMP SO-FS2	8.30E+00	2.00E+01	1.51E+01	2.77E+00
	VSMP SO	3.43E+00	<b>5.15E+00</b>	<b>4.03E+00(+)</b>	<b>4.68E−01</b>
F4: Griewank	SMP SO-RS	4.77E−01	1.06E+00	7.98E−01	<b>1.68E−01</b>
	SMP SO-AS	<b>1.59E−01</b>	<b>8.17E−01</b>	<b>4.98E−01(+)</b>	1.87E−01
	SMP SO-FS1	1.36E+00	6.22E+00	3.26E+00	1.14E+00
	SMP SO-FS2	1.71E+00	5.56E+00	3.33E+00	9.85E−01
	VSMP SO	6.50E−01	1.00E+00	8.54E−01	8.07E−02
F5: Rastrigin	SMP SO-RS	<b>3.72E+01</b>	1.62E+02	9.92E+01	2.80E+01
	SMP SO-AS	5.01E+01	<b>1.23E+02</b>	<b>8.80E+01(+)</b>	<b>1.69E+01</b>
	SMP SO-FS1	1.96E+02	3.22E+02	2.64E+02	2.96E+01
	SMP SO-FS2	2.22E+02	3.50E+02	2.95E+02	3.38E+01
	VSMP SO	2.01E+02	2.94E+02	2.51E+02	2.42E+01
F6 (F10 in CEC05)	SMP SO-RS	−2.70E+02	−8.34E+01	−2.09E+02	3.68E+01
	SMP SO-AS	− <b>2.71E+02</b>	− <b>1.69E+02</b>	− <b>2.18E+02(+)</b>	2.90E+01
	SMP SO-FS1	−1.23E+01	1.24E+02	6.03E+01	3.61E+01
	SMP SO-FS2	−6.45E+01	1.37E+02	4.99E+01	4.97E+01
	VSMP SO	−8.90E+01	1.26E+01	−4.76E+01	<b>2.54E+01</b>
F7 (F19 in CEC05)	SMP SO-RS	9.55E+02	1.09E+03	1.01E+03	3.67E+01
	SMP SO-AS	9.68E+02	1.10E+03	1.03E+03	3.81E+01
	SMP SO-FS1	1.00E+03	1.23E+03	1.11E+03	5.95E+01
	SMP SO-FS2	1.01E+03	1.22E+03	1.13E+03	5.04E+01
	VSMP SO	<b>9.26E+02</b>	<b>9.97E+02</b>	<b>9.43E+02(+)</b>	<b>1.51E+01</b>

promising solutions from the current population and the current RBF model to evaluation. This differs from the strategy of all SMP SO algorithms that only searches to evaluate the most promising solution from the current population. From Tables 12, 13 and 15, the results of VSMP SO are significantly different from those of other algorithms. On 30D functions, VSMP SO obtained the best results on F1, F2, F3 and F7, besides obtaining slightly lower results with SMP SO-RS on 30-D F4, F5 and F6. However, on 50D functions, VSMP SO achieved better optimization results on F1, F2, F3, F4 and F7, slightly less than SMP SO-RS on F4 and

F5. Furthermore, on 100D functions, VSMP SO achieved the best optimization results on F1, F2, F3, F4, F5 and F7, besides obtaining slightly lower results with SMP SO-AS on F6. VSMP SO has clear advantages on 100D than 50D and 30D, the improvement becomes more remarkable when the dimension of decision space D increases.

From the results mentioned above, we concluded that combining the two main innovations of the proposed VSMP SO in “Proposed VSMP SO algorithm” contributed to improving the performance of the proposed algorithm. First, from the results of comparing SMP SO-RS and three other algorithms

**Table 13** Statistical results of VSMP SO-RS, VSMP SO-AS, VSMP SO-FS1, VSMP SO-FS2 and VSMP SO on 50-D F1–F7 with 550 FEs

Function	Algorithm	Best	Worst	Mean (Friedman test)	SD
F1: Ellipsoid	SMP SO-RS	2.29E+01	3.46E+02	1.13E+02	7.43E+01
	SMP SO-AS	1.24E+01	2.98E+02	8.49E+01	6.20E+01
	SMP SO-FS1	1.49E+02	9.16E+02	3.34E+02	1.74E+02
	SMP SO-FS2	1.50E+02	1.01E+03	3.37E+02	1.79E+02
	VSMP SO	<b>3.79E+00</b>	<b>2.53E+01</b>	<b>1.19E+01(+)</b>	<b>6.13E+00</b>
F2: Rosenbrock	SMP SO-RS	2.17E+02	4.82E+02	3.25E+02	5.69E+01
	SMP SO-AS	2.27E+02	4.45E+02	3.31E+02	4.94E+01
	SMP SO-FS1	3.43E+02	1.82E+03	9.13E+02	3.84E+02
	SMP SO-FS2	3.34E+02	2.81E+03	1.05E+03	6.04E+02
	VSMP SO	<b>1.14E+02</b>	<b>2.75E+02</b>	<b>1.88E+02(+)</b>	<b>3.04E+01</b>
F3: Ackley	SMP SO-RS	9.13E+00	<b>1.64E+01</b>	1.35E+01	<b>1.79E+00</b>
	SMP SO-AS	5.63E+00	1.73E+01	1.29E+01	2.77E+00
	SMP SO-FS1	1.72E+01	2.04E+01	1.91E+01	<b>6.51E−01</b>
	SMP SO-FS2	1.71E+01	2.04E+01	1.89E+01	7.75E−01
	VSMP SO	<b>4.09E+00</b>	2.00E+01	<b>8.68E+00(+)</b>	6.36E+00
F4: Griewank	SMP SO-RS	1.07E+00	3.87E+01	4.00E+00	7.08E+00
	SMP SO-AS	1.14E+00	1.93E+01	4.25E+00	4.52E+00
	SMP SO-FS1	8.17E+00	1.15E+02	2.48E+01	2.28E+01
	SMP SO-FS2	8.25E+00	5.26E+01	2.22E+01	1.14E+01
	VSMP SO	<b>6.42E−01</b>	<b>9.84E−01</b>	<b>8.34E−01(+)</b>	<b>7.38E−02</b>
F5: Rastrigin	SMP SO-RS	1.32E+02	<b>2.75E+02</b>	2.09E+02	<b>3.57E+01</b>
	SMP SO-AS	<b>1.20E+02</b>	2.76E+02	<b>1.92E+02(+)</b>	3.92E+01
	SMP SO-FS1	4.28E+02	6.32E+02	5.32E+02	4.54E+01
	SMP SO-FS2	4.79E+02	6.22E+02	5.41E+02	3.94E+01
	VSMP SO	2.50E+02	4.81E+02	4.04E+02	5.97E+01
F6 (F10 in CEC05)	SMP SO-RS	− 1.96E+02	− <b>3.90E+01</b>	− <b>1.23E+02(+)</b>	4.42E+01
	SMP SO-AS	− <b>2.11E+02</b>	− 4.39E+00	− 1.10E+02	5.26E+01
	SMP SO-FS1	1.89E+02	3.68E+02	2.80E+02	<b>4.28E+01</b>
	SMP SO-FS2	1.90E+02	3.63E+02	2.67E+02	4.82E+01
	VSMP SO	− 1.15E+01	1.87E+02	1.05E+02	5.72E+01
F7 (F19 in CEC05)	SMP SO-RS	1.02E+03	1.15E+03	1.08E+03	3.69E+01
	SMP SO-AS	1.05E+03	1.23E+03	1.10E+03	3.97E+01
	SMP SO-FS1	1.09E+03	1.34E+03	1.19E+03	6.13E+01
	SMP SO-FS2	1.07E+03	1.25E+03	1.17E+03	4.07E+01
	VSMP SO	<b>9.49E+02</b>	<b>1.04E+03</b>	<b>9.81E+02(+)</b>	<b>2.49E+01</b>

(SMP SO-AS, SMP SO-FS1, and SMP SO-FS2), the random sample selection method is the same as the one we used in VSMP SO, eliminates the weaknesses of the fixed samples. Second, comparing the results of SMP SO-RS and VSMP SO, looking for the most promising solution from two different angles, especially the most promising solution from the current RBF model, quickening the optimal speed.

### Parameter sensitivity analysis

The parameters in VSMP SO, like the parameters of the optimizer SLPSO and the number of consumed function evaluations (FEs), may significantly influence the proposed algorithm's performance. For a fair comparison, the parameters of the optimizer SLPSO are set the same as the recommended parameters in [40]. For all the algorithms compared in this paper, the termination condition depends on the number of consumed function evaluations (FEs). The computational budget is less than  $11 \cdot D$  number of func-

**Table 14** Statistical results (best fitness and cost time) obtained by SMPSO-RS, SMPSO-AS, SMPSO-FS1, SMPSO-FS2 and VSMPSO on 30-, 50-, 100-D F1–F7 with 11 · D FEs

Fitness/time	Dimension	VSMPSO	SMPSO-RS	SMPSO-AS	SMPSO-FS1	SMPSO-FS2
F1: Ellipsoid	30D	2.33E+00/5.45E+01	8.84E+00/6.78E+00	4.21E+00/8.12E+00	4.77E+01/8.12E+00	3.68E+01/5.60E+00
	50D	1.19E+01/1.72E+02	1.13E+02/2.52E+01	8.49E+01/2.73E+01	3.34E+02/2.61E+01	3.37E+02/2.31E+01
	100D	3.60E+01/8.71E+02	7.69E+01/6.08E+02	7.31E+01/2.57E+02	4.13E+02/6.06E+02	3.52E+02/1.87E+02
F2: Rosenbrock	30D	1.12E+02/5.43E+01	1.52E+02/6.86E+00	1.38E+02/8.28E+00	3.01E+02/6.72E+00	2.62E+02/5.85E+00
	50D	1.88E+02/1.61E+02	3.25E+02/2.50E+01	3.31E+02/3.07E+01	9.13E+02/2.05E+01	1.05E+03/2.03E+01
	100D	1.59E+02/8.24E+02	1.88E+02/6.07E+02	1.60E+02/2.54E+02	4.69E+02/5.76E+02	4.06E+02/2.00E+02
F3: Ackley	30D	4.03E+00/5.52E+01	5.43E+00/6.95E+00	6.07E+00/8.16E+00	1.49E+01/5.90E+00	1.51E+01/5.76E+00
	50D	8.68E+00/1.81E+02	1.35E+01/5.76E+00	1.29E+01/3.12E+01	1.91E+01/3.12E+01	1.89E+01/2.56E+01
	100D	1.06E+01/8.33E+02	1.41E+01/6.21E+02	1.50E+01/2.74E+02	1.65E+01/5.98E+02	1.67E+01/2.06E+02
F4: Griewank	30D	8.54E−01/5.64E+01	7.98E−01/9.30E+00	4.98E−01/1.09E+01	3.26E+00/8.83E+00	3.33E+00/1.00E+01
	50D	8.34E−01/1.80E+02	4.00E+00/3.03E+01	4.25E+00/3.21E+01	2.48E+01/3.17E+01	2.22E+01/3.02E+01
	100D	8.17E−01/8.49E+02	1.93E+00/6.23E+02	1.87E+00/2.75E+02	5.74E+00/6.22E+02	6.11E+00/2.03E+02
F5: Rastrigin	30D	9.92E+01/5.52E+01	2.51E+02/6.90E+00	8.80E+01/8.13E+00	2.95E+02/5.51E+00	2.95E+02/5.50E+00
	50D	2.09E+02/1.74E+02	2.50E+02/2.50E+01	1.92E+02/3.18E+01	5.32E+02/2.60E+01	5.41E+02/2.32E+01
	100D	3.43E+02/8.80E+02	4.79E+02/6.08E+02	2.79E+02/2.52E+02	8.86E+02/5.97E+02	8.91E+02/1.86E+02
F6 (F10 in CEC05)	30D	−2.09E+02/5.53E+01	−4.76E+01/5.85E+00	−2.18E+00/6.89E+00	6.03E+01/7.29E+00	4.99E+01/5.99E+00
	50D	1.05E+02/1.76E+02	−1.23E+02/2.22E+01	−1.10E+02/2.73E+01	2.80E+02/2.66E+01	2.67E+02/2.11E+01
	100D	7.28E+02/9.45E+02	9.40E+02/7.16E+02	1.32E+03/2.92E+02	1.21E+03/6.57E+02	1.14E+03/2.38E+02
F7 (F19 in CEC05)	30D	1.01E+03/5.58E+01	9.43E+02/6.28E+00	1.03E+03/7.73E+00	1.11E+03/8.12E+00	1.13E+03/6.79E+00
	50D	9.81E+02/1.77E+02	1.08E+03/2.41E+01	1.10E+03/2.90E+01	1.19E+03/2.81E+01	1.17E+03/2.21E+01
	100D	1.40E+03/1.11E+03	1.41E+03/1.08E+03	1.46E+03/4.70E+02	1.46E+03/1.00E+03	1.45E+03/4.01E+02
All/NDS/DS	30D	7/4/3	7/7/0	7/5/2	7/4/3	7/4/3
	50D	7/5/2	7/7/0	7/3/4	7/2/5	7/7/0
	100D	7/6/1	7/3/4	7/5/2	7/0/7	7/7/0

**Table 15** Statistical results of SMPSO-RS, SMPSO-AS, SMPSO-FS1, SMPSO-FS2, VSMPSO on 100-D with 1100 FEs

Function	Algorithm	Best	Worst	Mean	SD
F1: Ellipsoid	SMPSO-RS	3.04E+01	1.33E+02	7.69E+01	2.70E+01
	SMPSO-AS	2.65E+01	1.37E+02	7.31E+01	2.59E+01
	SMPSO-FS1	1.68E+02	9.51E+02	4.13E+02	2.04E+02
	SMPSO-FS2	1.75E+02	6.29E+02	3.52E+02	9.56E+01
	VSMPSO	<b>2.17E+01</b>	<b>6.20E+01</b>	<b>3.60E+01(+)</b>	<b>9.31E+00</b>
F2: Rosenbrock	SMPSO-RS	1.53E+02	2.40E+02	1.88E+02	2.07E+01
	SMPSO-AS	<b>1.30E+02</b>	2.19E+02	1.60E+02	1.71E+01
	SMPSO-FS1	2.62E+02	1.25E+03	4.69E+02	1.90E+02
	SMPSO-FS2	2.67E+02	6.40E+02	4.06E+02	6.48E+01
	VSMPSO	1.31E+02	<b>2.09E+02</b>	<b>1.59E+02(+)</b>	<b>1.54E+01</b>
F3: Ackley	SMPSO-RS	1.21E+01	1.56E+01	1.41E+01	8.14E−01
	SMPSO-AS	1.34E+01	1.58E+01	1.50E+01	5.31E−01
	SMPSO-FS1	1.55E+01	1.77E+01	1.65E+01	6.04E−01
	SMPSO-FS2	1.42E+01	1.83E+01	1.67E+01	8.70E−01
	VSMPSO	<b>8.24E+00</b>	<b>1.25E+01</b>	<b>1.06E+01(+)</b>	<b>8.99E−01</b>
F4: Griewank	SMPSO-RS	1.24E+00	3.10E+00	1.93E+00	5.01E−01
	SMPSO-AS	1.29E+00	3.65E+00	1.87E+00	5.54E−01
	SMPSO-FS1	3.45E+00	9.26E+00	5.74E+00	1.62E+00
	SMPSO-FS2	3.34E+00	1.07E+01	6.11E+00	1.80E+00
	VSMPSO	<b>6.58E−01</b>	<b>1.01E+00</b>	<b>8.17E−01(+)</b>	<b>8.24E−02</b>
F5: Rastrigin	SMPSO-RS	2.79E+02	4.29E+02	3.43E+02	5.20E+01
	SMPSO-AS	<b>2.06E+02</b>	<b>3.70E+02</b>	<b>2.79E+02(+)</b>	<b>4.07E+01</b>
	SMPSO-FS1	8.29E+02	9.49E+02	8.86E+02	3.17E+01
	SMPSO-FS2	8.26E+02	9.91E+02	8.91E+02	3.77E+01
	VSMPSO	3.09E+02	7.32E+02	4.79E+02	8.71E+01
F6 (F10 in CEC05)	SMPSO-RS	7.54E+02	1.36E+03	9.40E+02	1.24E+02
	SMPSO-AS	8.47E+02	2.68E+03	1.32E+03	4.95E+02
	SMPSO-FS1	8.54E+02	1.63E+03	1.21E+03	2.00E+02
	SMPSO-FS2	8.05E+02	1.73E+03	1.14E+03	2.06E+02
	VSMPSO	<b>5.86E+02</b>	<b>8.36E+02</b>	<b>7.28E+02(+)</b>	<b>6.55E+01</b>
F7 (F19 in CEC05)	SMPSO-RS	1.35E+03	1.47E+03	1.41E+03	3.13E+01
	SMPSO-AS	1.37E+03	1.54E+03	1.46E+03	4.14E+01
	SMPSO-FS1	1.36E+03	1.57E+03	1.46E+03	4.80E+01
	SMPSO-FS2	1.38E+03	1.53E+03	1.45E+03	4.26E+01
	VSMPSO	<b>1.33E+03</b>	<b>1.46E+03</b>	<b>1.40E+03(+)</b>	<b>4.05E+01</b>

tion evaluations (NFEs) [32], which means that the limited number of fitness evaluations is 11 times the dimension of the problem. In VSMPSO, the value of  $\lambda$  is 80% of the total sample number. Then, we only need to analyze the parameters  $\lambda$  of VSMPSO, to explore the influence of different sample sizes on optimization performance. In the next experiment, we analyzed the performance comparison of VSMPSO for different sample sizes. The comparison algorithms VSMPSO-50, VSMPSO-60, VSMPSO-70, VSMPSO-80, VSMPSO-90, and VSMPSO-100 represent VSMPSO with different samples, which respectively represent the number of selected samples  $\lambda$  accounting for 50%,

60%, 70%, 80%, 90% and 100% of the total sample size, and VSMPSO-100 means all samples for modeling. Furthermore, all parameters in VSMPSO-80 are the same as those in VSMPSO, meaning the results of VSMPSO-80 are consistent with VSMPSO mentioned in the abovementioned experiments.

From Figs. 18, 19, 20, 21, 22, 23 and Tables 16, 17, 18, VSMPSO-80 performs best on 30D F1 and F3, 50D F3 and has no significant difference from VMPSO-100 on 100D functions. From Table 19 that, VSMPSO-80 obtained 3 non-dominant solutions on 30D functions, 4 non-dominant solutions on 50D functions, 5 non-dominant solutions on

**Table 16** Statistical results of VSMPSO with different samples on 30-D F1–F7 with 330 FEs

Function	Algorithm	Best	Worst	Mean (Friedman test)	SD	Meantime
F1: Ellipsoid	VSMPSO-50	2.20E+00	9.09E+00	5.60E+00	2.41E+00	5.61E+01
	VSMPSO-60	1.60E+00	5.17E+00	4.48E+00	1.40E+00	6.30E+01
	VSMPSO-70	1.71E+00	3.63E+00	3.17E+00	7.46E−01	6.99E+01
	VSMPSO-80	<b>4.45E−01</b>	<b>1.41E+00</b>	<b>7.69E−01(+)</b>	<b>4.65E−01</b>	7.64E+01
	VSMPSO-90	7.38E−01	2.89E+00	2.43E+00	7.21E−01	8.32E+01
	VSMPSO-100	5.41E−01	3.55E+00	1.56E+00	1.37E+00	8.93E+01
F2: Rosenbrock	SLPSO-50	1.16E+02	1.72E+02	1.47E+02	2.20E+01	6.24E+01
	VSMPSO-60	<b>6.69E+01</b>	1.78E+02	<b>9.75E+01(+)</b>	2.90E+01	7.10E+01
	VSMPSO-70	8.89E+01	<b>1.22E+02</b>	1.12E+02	1.35E+01	7.55E+01
	VSMPSO-80	1.03E+02	1.27E+02	1.14E+02	<b>7.61E+00</b>	8.23E+01
	VSMPSO-90	8.41E+01	1.40E+02	1.22E+02	1.63E+01	8.91E+01
	VSMPSO-100	7.17E+01	1.26E+02	1.11E+02	2.26E+01	9.56E+01
F3: Ackley	SLPSO-50	7.47E+00	2.00E+01	1.87E+01	3.81E+00	5.72E+01
	VSMPSO-60	4.82E+00	2.00E+01	7.86E+00	6.16E+00	6.37E+01
	VSMPSO-70	4.58E+00	4.85E+00	4.60E+00	6.72E−02	7.02E+01
	VSMPSO-80	4.01E+00	4.09E+00	4.02E+00	<b>1.87E−02</b>	7.69E+01
	VSMPSO-90	<b>3.62E+00</b>	<b>3.82E+00</b>	<b>3.63E+00(+)</b>	5.10E−02	8.37E+01
	VSMPSO-100	3.93E+00	5.17E+00	4.57E+00	3.82E−01	9.03E+01
F4: Griewank	SLPSO-50	9.50E−01	9.81E−01	9.59E−01	<b>1.40E−02</b>	5.70E+01
	VSMPSO-60	8.61E−01	9.66E−01	9.20E−01	5.30E−02	6.33E+01
	VSMPSO-70	8.38E−01	9.81E−01	9.21E−01	7.02E−02	7.05E+01
	VSMPSO-80	8.05E−01	8.55E−01	8.21E−01	1.76E−02	7.78E+01
	VSMPSO-90	7.13E−01	<b>8.05E−01</b>	7.99E−01	2.34E−02	8.37E+01
	VSMPSO-100	<b>4.99E−01</b>	8.40E−01	<b>7.19E−01(+)</b>	1.47E−01	9.02E+01
F5: Rastrigin	SLPSO-50	2.27E+02	2.69E+02	2.37E+02	1.81E+01	5.69E+01
	VSMPSO-60	2.33E+02	2.62E+02	2.50E+02	1.32E+01	6.41E+01
	VSMPSO-70	2.27E+02	2.84E+02	2.64E+02	1.92E+01	7.05E+01
	VSMPSO-80	2.42E+02	2.80E+02	2.58E+02	1.83E+01	7.76E+01
	VSMPSO-90	1.56E+02	2.52E+02	1.83E+02	4.00E+01	8.39E+01
	VSMPSO-100	<b>1.22E+02</b>	<b>1.58E+02</b>	<b>1.46E+02(+)</b>	<b>1.61E+01</b>	9.04E+01
F6 (F10 in CEC05)	SLPSO-50	−1.07E+02	−1.72E+01	−6.13E+01	2.08E+01	6.36E+01
	VSMPSO-60	−9.76E+01	− <b>1.90E+01</b>	−6.07E+01	2.39E+01	7.09E+01
	VSMPSO-70	−8.73E+01	−6.43E+00	−5.89E+01	<b>2.22E+01</b>	7.71E+01
	VSMPSO-80	−9.72E+01	−1.79E+01	−5.81E+01	2.26E+01	8.61E+01
	VSMPSO-90	−1.14E+02	2.63E+01	−4.48E+01	3.26E+01	9.13E+01
	VSMPSO-100	− <b>1.87E+02</b>	−1.66E+01	− <b>1.06E+02(+)</b>	4.96E+01	9.74E+01
F7 (F19 in CEC05)	SLPSO-50	9.32E+02	9.51E+02	9.44E+02	8.89E+00	6.11E+01
	VSMPSO-60	9.23E+02	1.01E+03	9.45E+02	1.90E+01	7.26E+01
	VSMPSO-70	9.23E+02	<b>9.39E+02</b>	<b>9.37E+02(+)</b>	<b>6.17E+00</b>	7.52E+01
	VSMPSO-80	<b>9.23E+02</b>	9.77E+02	9.40E+02 (≈)	1.37E+01	8.04E+01
	VSMPSO-90	9.31E+02	9.50E+02	9.47E+02	6.07E+00	8.90E+01
	VSMPSO-100	9.38E+02	9.58E+02	9.42E+02	7.22E+00	9.30E+01



**Table 17** Statistical results of VSMPSO with different samples on 50-D F1–F7 with 550 FEs

Function	Algorithm	Best	Worst	Mean	SD	Meantime
F1: Ellipsoid	VSMPSO-50	8.61E+00	<b>9.35E+00</b>	<b>8.96E+00(+)</b>	<b>1.97E−01</b>	1.63E+02
	VSMPSO-60	3.67E+00	2.14E+01	1.11E+01	3.73E+00	1.87E+02
	VSMPSO-70	4.56E+00	2.68E+01	1.16E+01	5.41E+00	2.10E+02
	VSMPSO-80	<b>3.26E+00</b>	3.43E+01	1.19E+01	7.18E+00	2.33E+02
	VSMPSO-90	3.41E+00	2.54E+01	1.06E+01	4.92E+00	2.56E+02
	VSMPSO-100	2.36E+00	2.62E+01	1.04E+01	6.38E+00	2.83E+02
F2: Rosenbrock	SLPSO-50	<b>8.78E+01</b>	1.74E+02	<b>1.33E+02(+)</b>	<b>1.91E+01</b>	1.71E+02
	VSMPSO-60	9.78E+01	<b>1.72E+02</b>	1.41E+02	1.97E+01	1.94E+02
	VSMPSO-70	1.33E+02	2.32E+02	1.70E+02	2.59E+01	2.16E+02
	VSMPSO-80	1.51E+02	2.74E+02	2.03E+02	3.33E+01	2.40E+02
	VSMPSO-90	1.24E+02	2.93E+02	1.95E+02	3.74E+01	2.66E+02
	VSMPSO-100	1.26E+02	2.30E+02	1.62E+02	3.00E+01	2.86E+02
F3: Ackley	SLPSO-50	1.99E+01	2.00E+01	2.00E+01	5.91E−03	1.68E+02
	VSMPSO-60	1.90E+01	2.00E+01	1.99E+01	<b>1.71E−01</b>	1.90E+02
	VSMPSO-70	4.43E+00	2.00E+01	1.59E+01	6.35E+00	2.15E+02
	VSMPSO-80	4.05E+00	6.44E+00	<b>4.58E+00(+)</b>	6.82E−01	2.34E+02
	VSMPSO-90	3.84E+00	<b>6.10E+00</b>	4.96E+00	7.12E−01	2.63E+02
	VSMPSO-100	<b>3.74E+00</b>	6.82E+00	5.31E+00	7.60E−01	2.86E+02
F4: Griewank	SLPSO-50	8.04E−01	1.02E+00	9.02E−01	<b>4.69E−02</b>	1.66E+02
	VSMPSO-60	8.00E−01	9.91E−01	9.02E−01	4.96E−02	1.90E+02
	VSMPSO-70	7.39E−01	1.01E+00	8.64E−01	6.02E−02	2.13E+02
	VSMPSO-80	6.42E−01	1.02E+00	8.24E−01	9.82E−02	2.35E+02
	VSMPSO-90	6.05E−01	<b>9.19E−01</b>	7.41E−01	6.39E−02	2.62E+02
	VSMPSO-100	<b>5.32E−01</b>	9.73E−01	<b>6.90E−01(+)</b>	1.21E−01	2.87E+02
F5: Rastrigin	SLPSO-50	3.86E+02	5.07E+02	4.54E+02	3.25E+01	1.73E+02
	VSMPSO-60	3.89E+02	5.05E+02	4.50E+02	<b>2.94E+01</b>	1.96E+02
	VSMPSO-70	3.60E+02	4.91E+02	4.28E+02	3.59E+01	2.21E+02
	VSMPSO-80	2.80E+02	5.31E+02	3.88E+02	7.04E+01	2.42E+02
	VSMPSO-90	1.87E+02	4.33E+02	3.12E+02	6.34E+01	2.65E+02
	VSMPSO-100	<b>1.04E+02</b>	<b>3.01E+02</b>	<b>1.77E+02(+)</b>	4.41E+01	2.92E+02
F6 (F10 in CEC05)	SLPSO-50	8.72E+01	2.01E+02	1.36E+02	2.88E+01	1.74E+02
	VSMPSO-60	9.48E+01	2.16E+02	1.50E+02	<b>2.73E+01</b>	2.00E+02
	VSMPSO-70	8.00E+01	1.94E+02	1.37E+02	2.74E+01	2.20E+02
	VSMPSO-80	7.28E+01	2.29E+02	1.57E+02	4.41E+01	2.40E+02
	VSMPSO-90	−6.06E+01	1.99E+02	1.10E+02	5.68E+01	2.66E+02
	SLPSO-100	− <b>9.42E+01</b>	<b>7.01E+01</b>	− <b>4.33E+01(+)</b>	4.72E+01	2.89E+02
F7 (F19 in CEC05)	SLPSO-50	9.71E+02	9.94E+02	9.82E+02	1.15E+01	1.73E+02
	VSMPSO-60	9.56E+02	1.04E+03	9.84E+02	2.22E+01	1.96E+02
	VSMPSO-70	9.58E+02	1.03E+03	9.89E+02	2.06E+01	2.19E+02
	VSMPSO-80	9.73E+02	9.93E+02	9.75E+02	3.52E+00	2.39E+02
	VSMPSO-90	9.54E+02	<b>9.79E+02</b>	<b>9.66E+02(+)</b>	9.52E+00	2.64E+02
	VSMPSO-100	<b>9.53E+02</b>	1.05E+03	9.75E+02	<b>2.73E+01</b>	2.88E+02

**Table 18** Statistical results of VSMPSO with different samples on 100-D F1–F7 with 1100 FEs

Function	Algorithm	Best	Worst	Mean	SD	Meantime
F1: Ellipsoid	VSMPSO-50	4.02E+01	6.96E+01	5.60E+01	7.63E+00	7.30E+02
	VSMPSO-60	2.86E+01	8.16E+01	4.53E+01	9.55E+00	8.50E+02
	VSMPSO-70	2.95E+01	4.32E+01	4.18E+01	<b>2.43E+00</b>	9.76E+02
	VSMPSO-80	2.03E+01	5.68E+01	3.43E+01	1.00E+01	1.10E+03
	VSMPSO-90	2.19E+01	4.58E+01	3.40E+01	6.86E+00	1.23E+03
	VSMPSO-100	<b>1.87E+01</b>	<b>3.91E+01</b>	<b>2.14E+01(+)</b>	3.79E+00	1.37E+03
F2: Rosenbrock	VSMPSO-50	1.60E+02	2.22E+02	1.89E+02	1.81E+01	7.39E+02
	VSMPSO-60	1.56E+02	1.99E+02	1.80E+02	1.18E+01	8.59E+02
	VSMPSO-70	1.64E+02	2.01E+02	1.79E+02	<b>6.45E+00</b>	9.88E+02
	VSMPSO-80	1.48E+02	1.86E+02	1.67E+02	1.41E+01	1.11E+03
	VSMPSO-90	1.28E+02	1.70E+02	1.47E+02	1.24E+01	1.24E+03
	VSMPSO-100	<b>1.15E+02</b>	<b>1.44E+02</b>	<b>1.31E+02(+)</b>	8.27E+00	1.38E+03
F3: Ackley	VSMPSO-50	1.40E+01	1.74E+01	1.61E+01	6.52E−01	7.40E+02
	VSMPSO-60	1.10E+01	1.39E+01	1.27E+01	9.35E−01	8.53E+02
	VSMPSO-70	1.23E+01	1.37E+01	1.25E+01	<b>4.28E−01</b>	9.86E+02
	VSMPSO-80	8.77E+00	1.24E+01	1.02E+01(≈)	7.13E−01	1.11E+03
	VSMPSO-90	8.83E+00	<b>1.06E+01</b>	1.04E+01	4.81E−01	1.24E+03
	VSMPSO-100	<b>8.27E+00</b>	1.09E+01	<b>9.77E+00(+)</b>	6.56E−01	1.37E+03
F4: Griewank	VSMPSO-50	9.39E−01	1.08E+00	1.01E+00	4.30E−02	7.40E+02
	VSMPSO-60	8.82E−01	1.04E+00	9.62E−01	5.17E−02	8.60E+02
	VSMPSO-70	8.41E−01	1.02E+00	9.46E−01	<b>3.60E−02</b>	9.87E+02
	VSMPSO-80	6.58E−01	1.01E+00	8.38E−01	8.18E−02	1.11E+03
	VSMPSO-90	5.67E−01	<b>9.52E−01</b>	7.57E−01	1.00E−01	1.24E+03
	VSMPSO-100	<b>4.22E−01</b>	9.81E−01	<b>6.56E−01(+)</b>	1.17E−01	1.38E+03
F5: Rastrigin	VSMPSO-50	7.50E+02	9.83E+02	8.71E+02	7.05E+01	7.35E+02
	VSMPSO-60	7.10E+02	8.40E+02	7.82E+02	5.27E+01	8.52E+02
	VSMPSO-70	5.67E+02	6.30E+02	5.83E+02	<b>2.39E+01</b>	9.83E+02
	VSMPSO-80	3.98E+02	7.18E+02	5.31E+02	7.42E+01	1.11E+03
	VSMPSO-90	2.98E+02	4.91E+02	4.05E+02	5.08E+01	1.23E+03
	VSMPSO-100	<b>1.75E+02</b>	<b>4.32E+02</b>	<b>2.32E+02(+)</b>	5.21E+01	1.36E+03
F6 (F10 in CEC05)	VSMPSO-50	<b>6.32E+02</b>	7.26E+02	6.87E+02	3.40E+01	9.01E+02
	VSMPSO-60	6.21E+02	7.38E+02	<b>6.69E+02(+)</b>	3.97E+01	8.99E+02
	VSMPSO-70	6.35E+02	7.63E+02	6.96E+02	<b>3.60E+01</b>	1.03E+03
	VSMPSO-80	<b>6.03E+02</b>	7.72E+02	7.00E+02	4.31E+01	1.14E+03
	VSMPSO-90	6.67E+02	8.44E+02	7.44E+02	6.43E+01	1.28E+03
	VSMPSO-100	6.34E+02	2.50E+03	8.49E+02	3.52E+02	1.40E+03
F7 (F19 in CEC05)	VSMPSO-50	1.34E+03	1.38E+03	<b>1.36E+03(+)</b>	1.16E+01	1.22E+03
	VSMPSO-60	1.36E+03	1.41E+03	1.37E+03(≈)	8.52E+00	1.21E+03
	VSMPSO-70	1.35E+03	<b>1.36E+03</b>	1.36E+03(≈)	<b>2.54E+00</b>	1.33E+03
	VSMPSO-80	<b>1.32E+03</b>	1.46E+03	1.39E+03(≈)	3.58E+01	1.45E+03
	VSMPSO-90	1.33E+03	1.46E+03	1.37E+03(≈)	4.69E+01	1.58E+03
	VSMPSO-100	1.33E+03	1.43E+03	1.39E+03(≈)	1.66E+01	1.71E+03

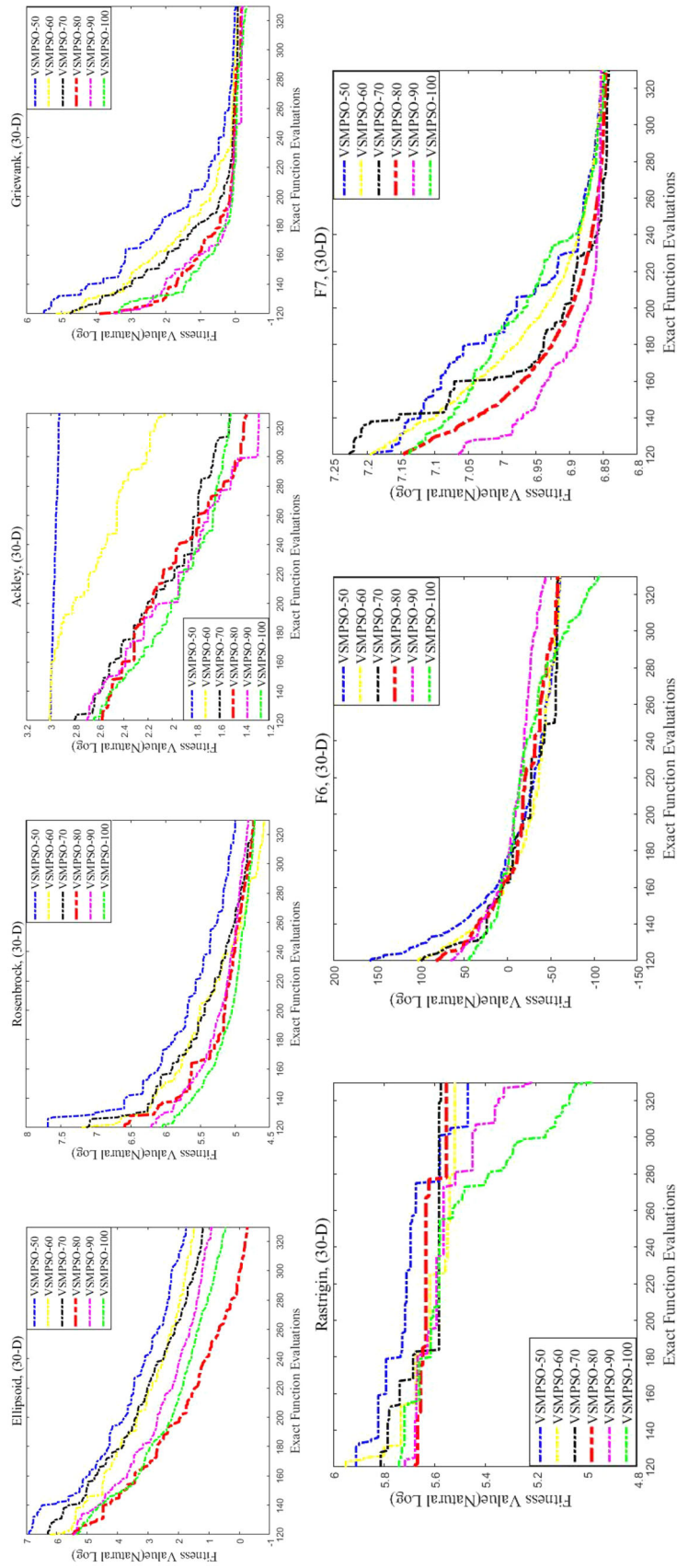


Fig. 18 Convergence profiles of VSMPSO with different samples on 30-D with 330 FEs

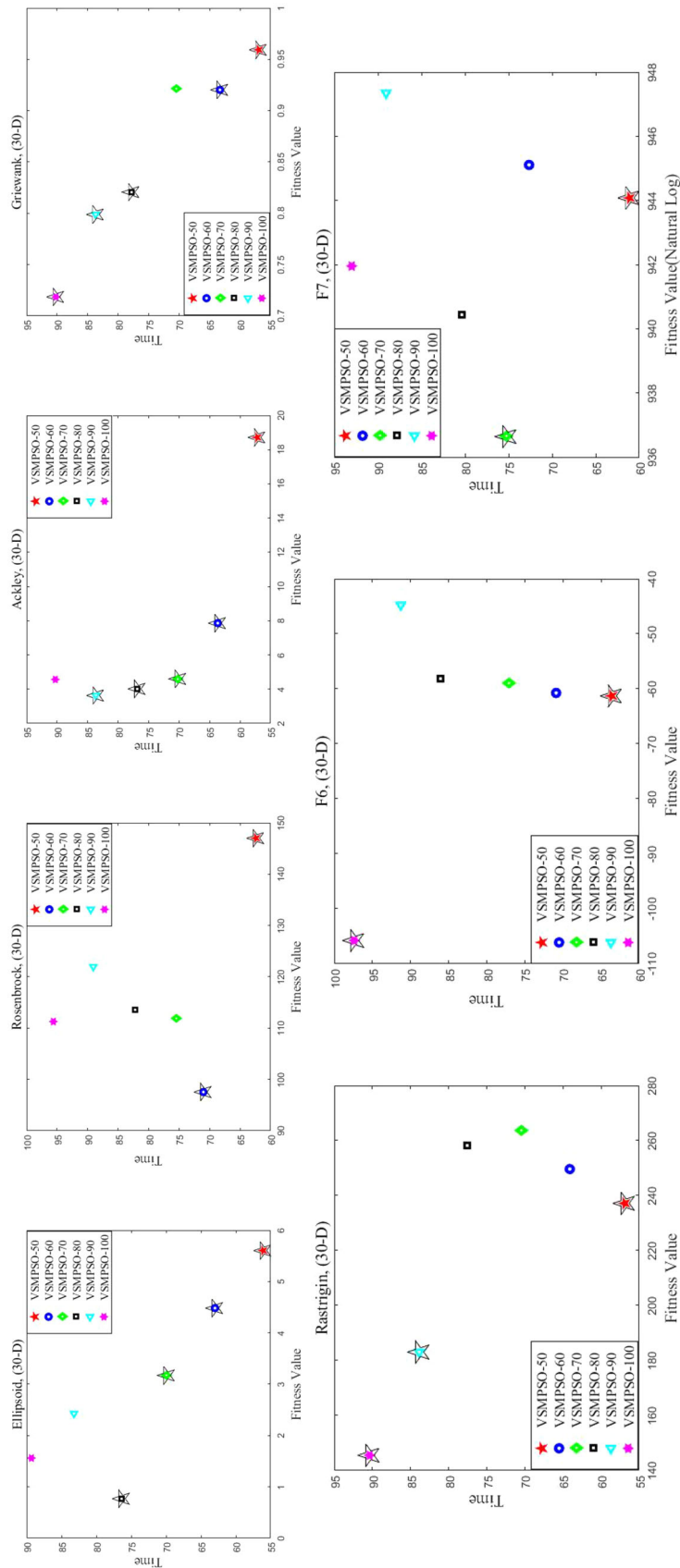


Fig. 19 Distribution of solutions for VSMPSO with different samples on 30-D F1–F7 with 330 FEs

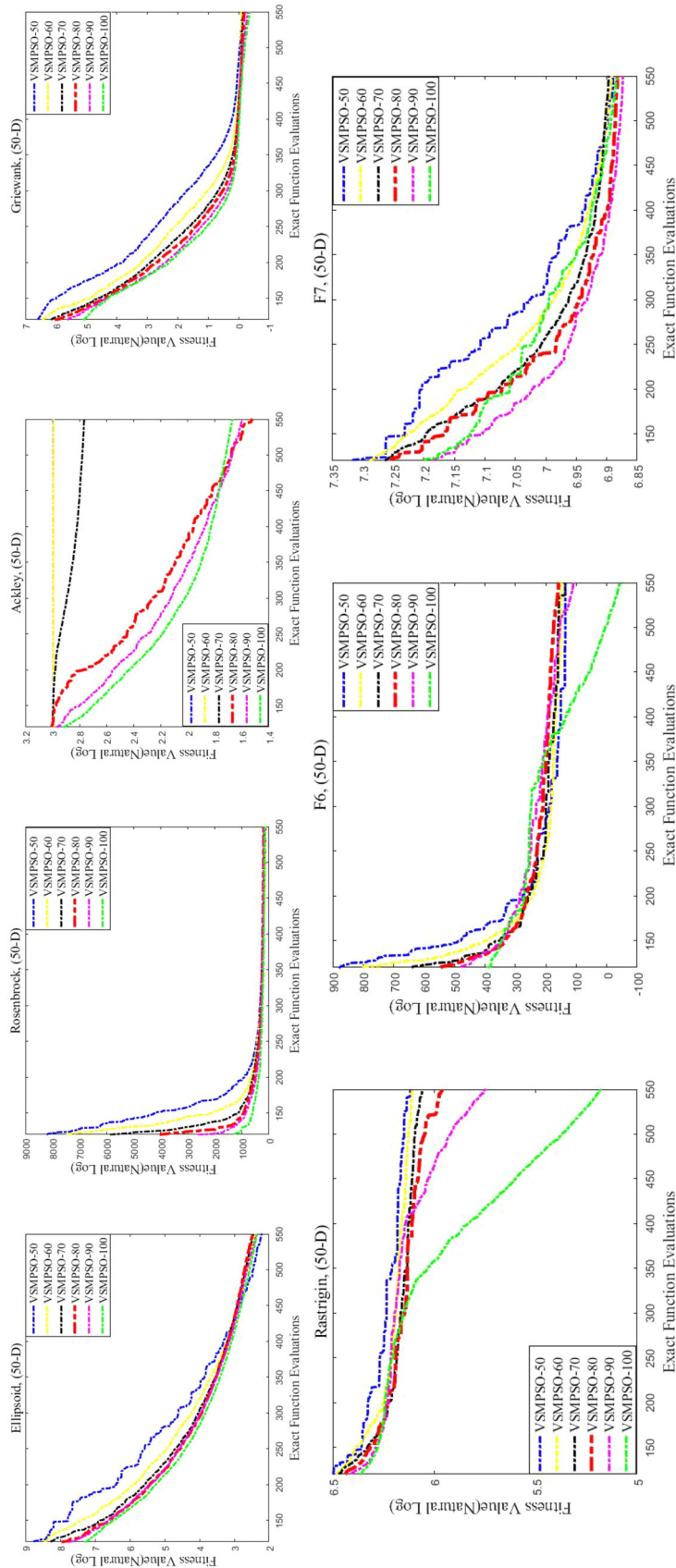


Fig. 20 Convergence profiles of VSMPSO with different samples on 50-D F1–F7 with 550 FEs

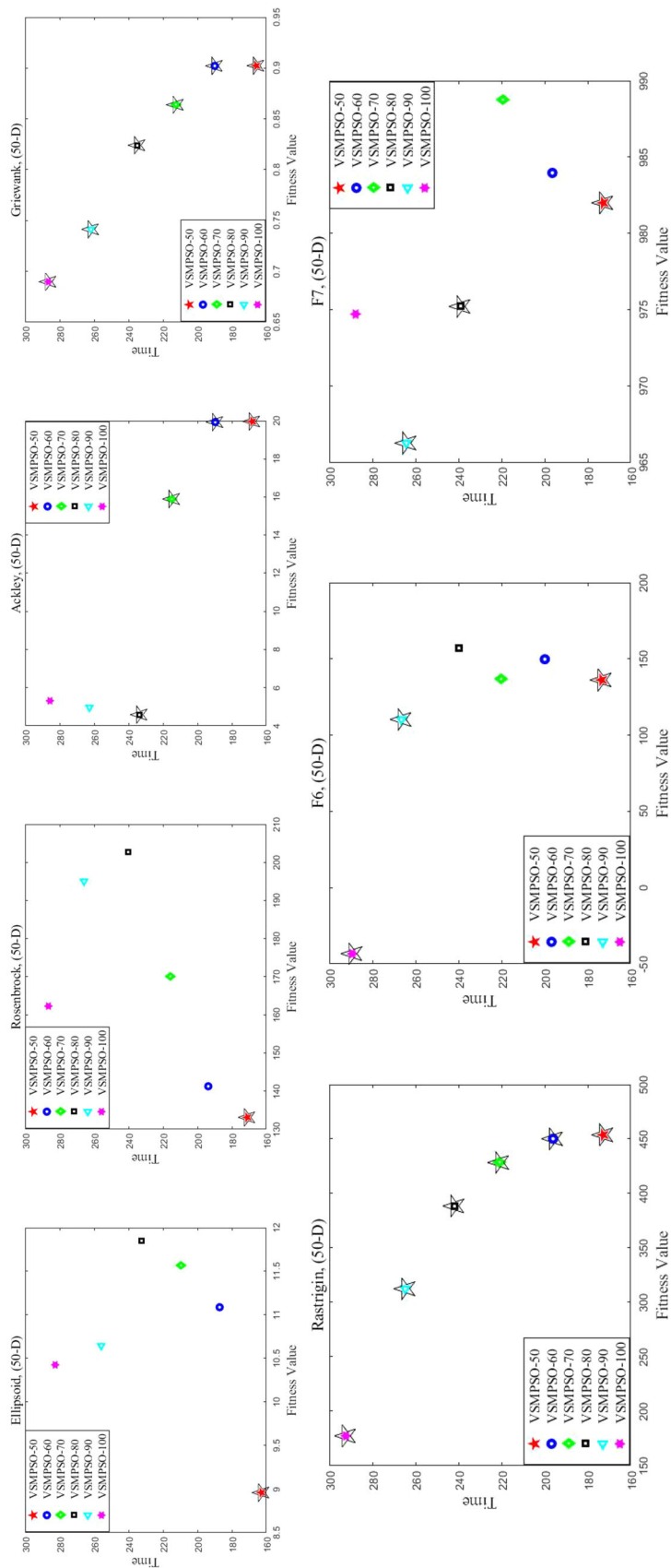


Fig. 21 Distribution of solutions for VSMPSO with different samples on 50-D F1–F7 with 550 FEs



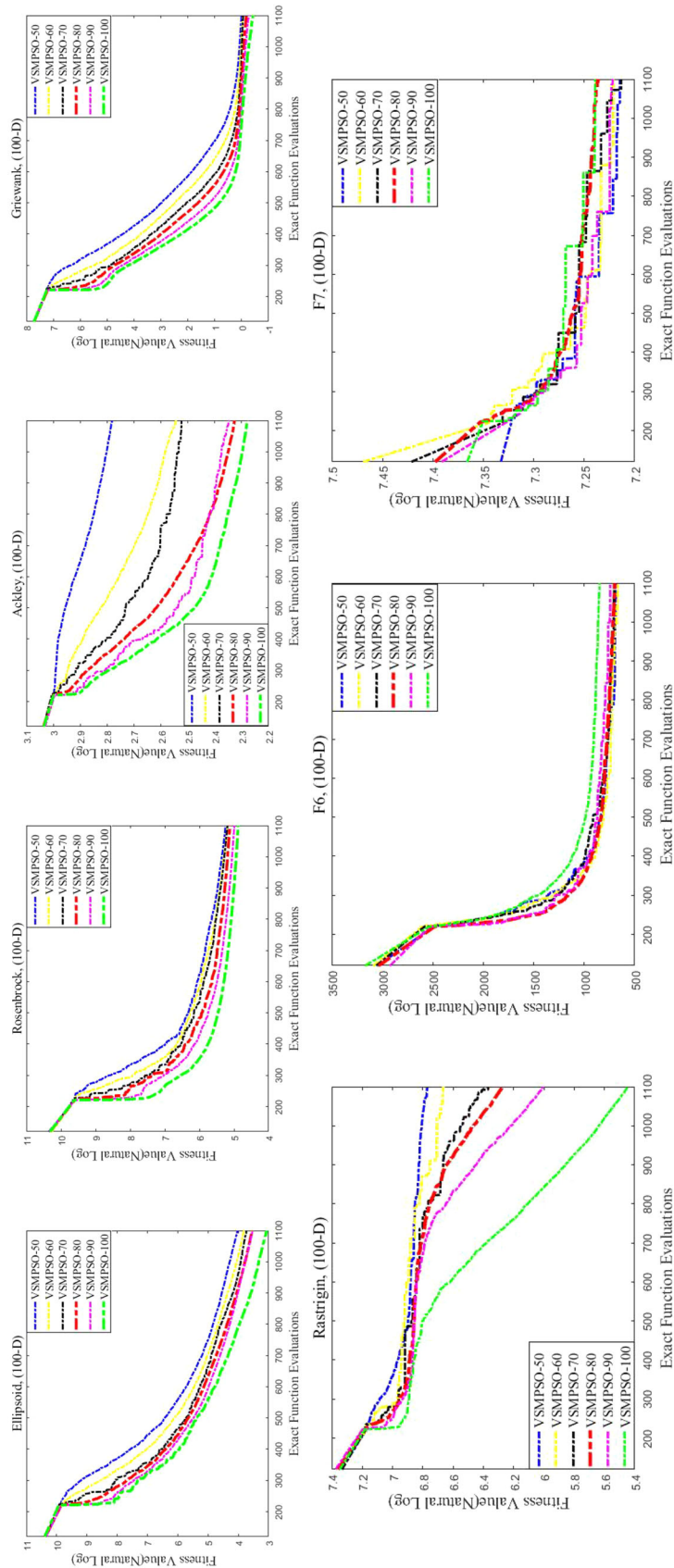


Fig. 22 Convergence profiles of VSMPSO with different samples on 100-D F1–F7 with 1100 FEs

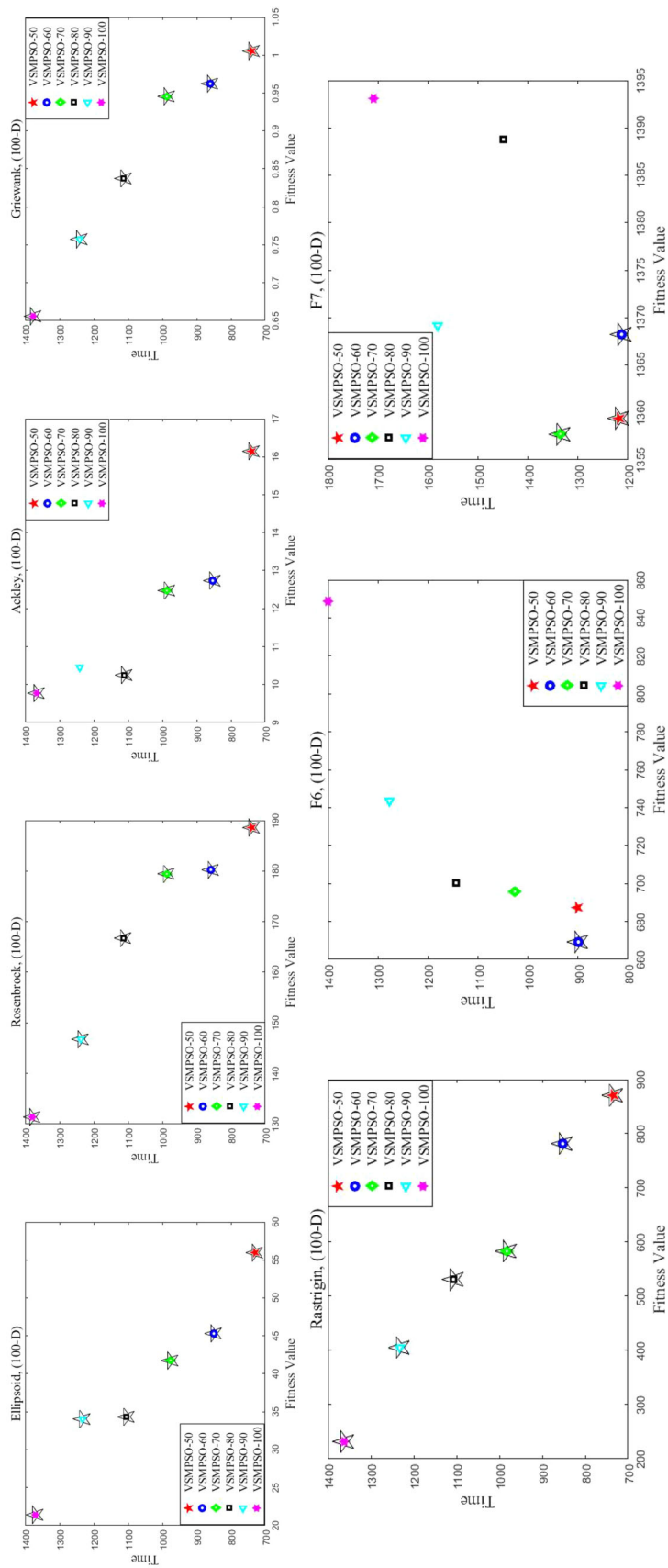


Fig. 23 Distribution of solutions of VSMPSO with different samples on 100-D F1–F7 with 1100 FEs

**Table 19** Obtained solutions of VSMPSO with different samples on 30-, 50-, 100-D F1–F7 with 11 · D FEs

Function	Dimension	VSMPSO-50	VSMPSO-60	VSMPSO-70	VSMPSO-80	VSMPSO-90	VSMPSO-100
F1 (Ellipsoid)	30D	5.60E+00/5.61E+01	4.48E+00/6.30E+01	3.17E+00/6.99E+01	7.69E−01/7.64E+01	2.43E+00/8.32E+01	1.56E+00/8.93E+01
	50D	8.96E+00/1.63E+02	1.11E+01/1.87E+02	1.16E+01/2.10E+02	1.19E+01/2.33E+02	1.06E+01/2.56E+02	1.04E+01/2.83E+02
	100D	5.60E+01/7.30E+02	4.53E+01/8.50E+02	4.18E+01/9.76E+02	3.43E+01/1.10E+03	3.40E+01/1.23E+03	2.14E+01/1.37E+03
F2 (Rosenbrock)	30D	1.47E+02/6.24E+01	9.75E+01/7.10E+01	1.12E+02/7.55E+01	1.14E+02/8.23E+01	1.22E+02/8.91E+01	1.11E+02/9.56E+01
	50D	1.33E+02/1.71E+02	1.41E+02/1.94E+02	1.70E+02/2.16E+02	2.03E+02/2.40E+02	1.95E+02/2.66E+02	1.62E+02/2.86E+02
	100D	1.89E+02/7.39E+02	1.80E+02/8.59E+02	1.79E+02/9.88E+02	1.67E+02/1.11E+03	1.47E+02/1.24E+03	1.31E+02/1.38E+03
F3 (Ackley)	30D	1.87E+01/5.72E+01	7.86E+00/6.37E+01	4.60E+00/7.02E+01	4.02E+00/7.69E+01	3.63E+00/8.37E+01	4.57E+00/9.03E+01
	50D	2.00E+01/1.68E+02	1.99E+01/1.90E+02	1.59E+01/2.15E+02	4.58E+00/2.34E+02	4.96E+00/2.63E+02	5.31E+00/2.86E+02
	100D	1.61E+01/7.40E+02	1.27E+01/8.53E+02	1.25E+01/9.86E+02	1.02E+01/1.11E+03	1.04E+01/1.24E+03	9.77E+00/1.37E+03
F4 (Griewank)	30D	9.59E−01/5.70E+01	9.20E−01/6.33E+01	9.21E−01/7.05E+01	8.21E−01/7.78+01	7.99E−01/8.37E+01	7.19E−01/9.02E+01
	50D	9.02E−01/1.66E+02	9.02E−01/1.90E+02	8.64E−01/2.13E+02	8.24E−01/2.35E+02	7.41E−01/2.62E+02	6.90E−01/2.87E+02
	100D	1.01E+00/7.40E+02	9.62E−01/8.60E+02	9.46E−01/9.87E+02	8.38E−01/1.11E+03	7.57E−01/1.24E+03	6.56E−01/1.38E+03
F5 (Rastrigin)	30D	2.37E+02/5.69E+01	2.50E+02/6.41E+01	2.64E+02/7.05E+01	2.58E+02/7.76E+01	1.83E+02/8.39E+01	1.46E+02/9.04E+01
	50D	4.54E+02/1.73E+02	4.50E+02/1.96E+02	4.28E+02/2.21E+02	3.88E+02/2.42E+02	3.12E+02/2.65E+02	1.77E+02/2.92E+02
	100D	8.71E+02/7.35E+02	7.82E+02/8.52E+02	5.83E+02/9.83E+02	5.31E+02/1.11E+03	4.05E+02/1.23E+03	2.32E+02/1.36E+03
F6 (F10 in CEC05)	30D	−6.13E+01/6.36E+01	−6.07E+01/7.09E+01	−5.89E+01/7.71E+01	−5.81E+01/8.61E+01	−4.48E+01/9.13E+01	−1.06E+02/9.74E+01
	50D	1.36E+02/1.74E+02	1.50E+02/2.00E+02	1.37E+02/2.20E+02	1.57E+02/2.40E+02	1.10E+02/2.66E+02	−4.33E+01/2.89E+02
	100D	6.87E+02/9.01E+02	6.69E+02/8.99E+02	6.96E+02/1.03E+03	7.00E+02/1.14E+03	7.44E+02/1.28E+03	8.49E+02/1.40E+03
F7 (F19 in CEC05)	30D	9.44E+02/6.11E+01	9.45E+02/7.26E+01	9.37E+02/7.52E+01	9.40E+02/8.04E+01	9.47E+02/8.90E+01	9.42E+02/9.30E+01
	50D	9.82E+02/1.73E+02	9.84E+02/1.96E+02	9.89E+02/2.19E+02	9.75E+02/2.39E+02	9.66E+02/2.64E+02	9.75E+02/2.88E+02
	100D	1.36E+03/1.22E+03	1.37E+03/1.21E+03	1.36E+03/1.33E+03	1.39E+03/1.45E+03	1.37E+03/1.58E+03	1.39E+03/1.71E+03
All/NDS/DS	30D	7/7/0	7/4/3	7/3/4	7/3/4	7/3/4	7/3/4
	50D	7/7/0	7/3/4	7/3/4	7/4/3	7/4/3	7/3/4
	100D	7/6/1	7/7/0	7/6/1	7/5/2	7/4/3	7/5/2

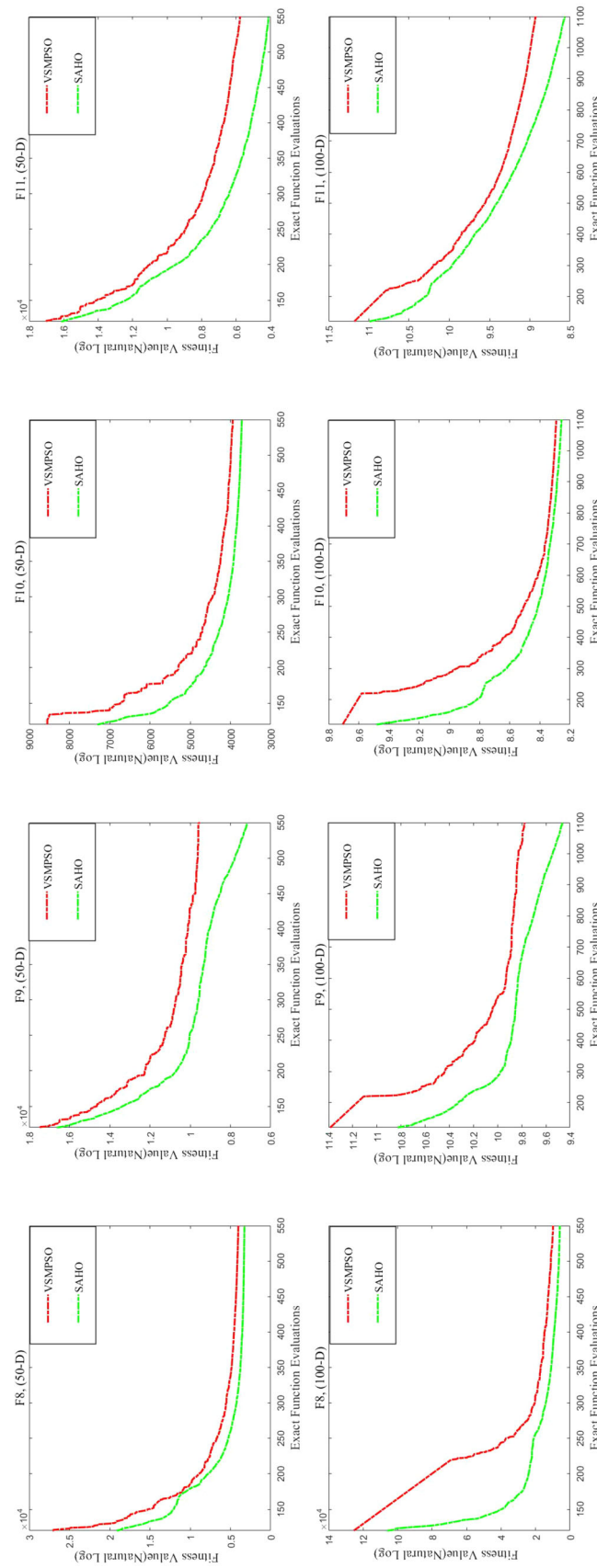


Fig. 24 Convergence profiles of SAHO and VSMPSO on 100-D F8–F11 with 1100 FEes

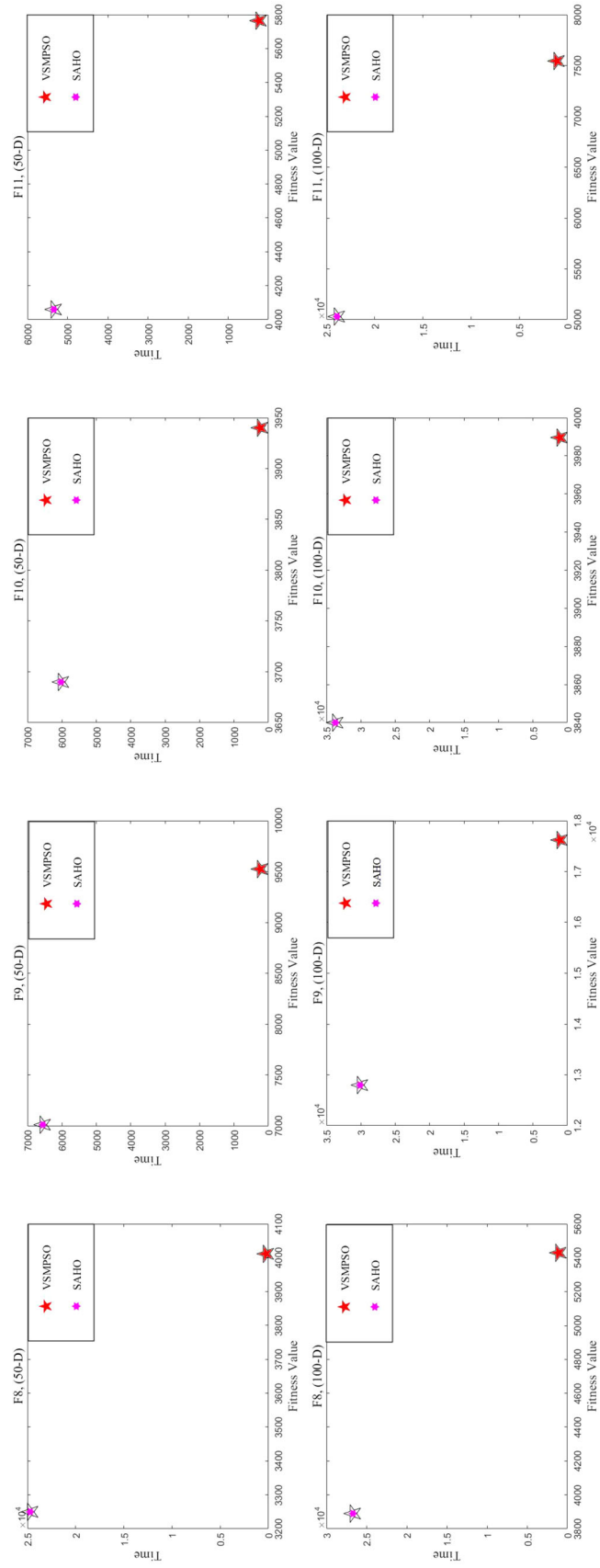


Fig. 25 Distribution of solutions of SAHO and VSMPSO on 100-D F8–F11 with 1100 FEs

**Table 20** Obtained solutions for VSMPSO and SAHO on 100-D F8–F11 with 1100 FEs

Mean/time	Dimension	VSMPSO-CEC17	SAHO-CEC17
F8	50D	4.01E+03/2.38E+02	3.25E+03/2.47E+04
	100D	5.43E+03/1.10E+03	3.89E+03/2.67E+04
F9	50D	9.53E+03/2.38E+02	7.01E+03/6.55E+03
	100D	1.76E+04/1.11E+03	1.28E+04/3.01E+04
F10	50D	3.94E+03/2.39E+02	3.69E+03/6.02E+03
	100D	3.99E+03/1.11E+03	3.84E+03/3.37E+04
F11	50D	5.76E+03/2.39E+02	4.06E+03/5.34E+03
	100D	7.54E+03/1.11E+03	5.03E+03/2.39E+04
All/NDS/DS	50D	7/7/0	7/7/0

100D functions. From Figs. 19, 21 and 23, most non-dominant solutions of VSMPSO-80 are basically knee points. Based on the aforementioned results, in general, the larger the sample size taken by VSMPSO, the better is the algorithm performance except F6. However, in terms of the time spent by the VSMPSO in different sample sizes, the larger the sample size, the longer is the running time of VSMPSO. By making a compromise between optimization performance and time-consumption, we select 80% of the total sample number as the number of training samples for modelling in VSMPSO.

### Numerical results on complex problems

To further compare algorithm performance, we compared VSMPSO with SAHO, which has the best optimization results from the previous comparison experiment, for test suite CEC 2017 [48], which has been recently proposed and is relatively complex. As can be seen from Fig. 24 curve convergence diagram, the performance of VSMPSO on F9 and F11 with 50D was slightly worse than SAHO; however, on other functions, especially on all 100D functions, VSMPSO obtained similar convergence curves to SAHO. However, from Fig. 25 and Table 20, the average time spent by SAHO is dozens or even hundreds of times that of VSMPSO. However, there is minimal difference between the optimal solutions obtained by VSMPSO and SAHO. It follows that, even on CEC 2017, the more complex benchmark functions, VSMPSO can achieve a better balance between the optimization effect and the time consumption, and obtains better optimization results on 200D dimensional problems.

### Conclusions

In this paper, a single surrogate-assisted evolutionary algorithm, called VSMPSO, has been proposed for high-dimensional expensive optimization problems. We have considered both optimization results and optimization time consumption as

bi-objectives when comparing algorithm performance. The proposed VSMPSO has shown promising performance on high-dimensional test problems with dimensions up to 200. It overcomes the shortcoming of using a single model in SAEAs, trapping in local optimum easily, and saves on training model time while improving on performance. Experimental results show that VSMPSO performed well on high-dimensional problems. In the future, we are interested in improving the performance of the proposed algorithm by considering the relationship between the candidate solutions and surrogate-management strategies and then extending it to higher-dimensional or multi-objective optimization problems.

**Acknowledgements** This research was partially supported by the National Science Foundation of China under Grant 62006143; the State Scholarship Fund of China Scholarship Council (202108370181); the National Science Foundation of Shandong Province (ZR2020MF152); the Special Fund Plan for Local Science and Technology Development, which is led by the central authority for major basic research projects in Shandong (ZR2018ZB0419); the Grant from the high-level scientific research project (2020RCYJ20) and in conjunction with the Big Data Analysis and Application Team (0209201904) in Shandong Women University.

### Declarations

**Conflict of interest** We should emphasize that there is no known conflict of interest in this publication, and there is no significant financial support that can affect its results. We confirm that the manuscript is our own village and has been recognized by all the designated authors. No one else meets the standards of the author, but it is not listed. We confirm that we have provided a valid e-mail address that the author can visit, which was set up to receive e-mail from [tianjie1023@outlook.com](mailto:tianjie1023@outlook.com).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the



permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Zhou Y, Jin Y, Ding J (2020) Surrogate-assisted evolutionary search of spiking neural architectures in liquid state machines. *Neurocomputing* 406:12–23
- Regis RG (2014) Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions. *IEEE Trans Evol Comput* 18:326–347
- Horner A, Beauchamp J, Haken L (1993) Machine tongues XVI: genetic algorithms and their application to FM matching synthesis. *Comput Music J* 17:17
- Fleming PJ, Purshouse RC (2002) Evolutionary algorithms in control systems engineering: a survey. *Control Eng Pract* 10:1223–1241
- Jin Y, Sendhoff B (2009) A systems approach to evolutionary multiobjective structural optimization and beyond. *IEEE Comput Intell Mag* 4:62–76
- Koziel S, Yang X-S (2011) *Computational optimization, methods and algorithms*. Springer, London
- Chugh T, Jin Y, Miettinen K, Hakanen J, Sindhya K (2018) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Trans Evol Comput* 22:129–142
- Liu H, Cai J, Ong Y-S (2017) An adaptive sampling approach for kriging metamodeling by maximizing expected prediction error. *Comput Chem Eng* 106:171–182
- Liu B, Zhang Q, Gielen GGE (2014) A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans Evol Comput* 18:180–192
- Dennis JE, Torczon V (1997) Managing approximation models in optimization. *Multidiscip Desi Optim State Art* 5:330–347
- Paenke I, Branke J, Jin Y (2006) Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Trans Evol Comput* 10:405–420
- Lian Y, Liou M-S (2005) Multiobjective optimization using coupled response surface model and evolutionary algorithm. *AIAA J* 43:1316–1325
- Ferrari S, Stengel RF (2005) Smooth function approximation using neural networks. *IEEE Trans Neural Netw* 16:24–38
- Amaritsakul Y, Chao C-K, Lin J (2013) Multiobjective optimization design of spinal pedicle screws using neural networks and genetic algorithm: mathematical models and mechanical validation. *Comput Math Methods Med* 1–9:2013
- Zhou Z, Ong YS, Nair PB, Keane AJ, Lum KY (2007) Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 37:66–76
- Gonzalez J, Rojas I, Ortega J, Pomares H, Fernandez J, Diaz A (2003) Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Trans Neural Netw* 14:1478–1495
- Sun C, Jin Y, Zeng J, Yang Yu (2014) A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Comput* 19:1461–1475
- Wang Y, Yin D-Q, Yang S, Sun G (2019) Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints. *IEEE Trans Cybern* 49:1642–1656
- Jin Y, Wang H, Chugh T, Guo D, Miettinen K (2019) Data-driven evolutionary optimization: an overview and case studies. *IEEE Trans Evol Comput* 23:442–458
- Wang X, Wang GG, Song B, Wang P, Wang Y (2019) A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. *IEEE Trans Evol Comput* 23:815–827
- Rana S, Li C, Gupta S, Nguyen V, Venkatesh S (2017) High dimensional Bayesian optimization with elastic Gaussian process. In: *Proceedings of the 34th international conference on machine learning*, volume 70 of *proceedings of machine learning research*. PMLR, 06–11 Aug, pp 2883–2891
- Li J-Y, Zhan Z-H, Wang H, Zhang J (2021) Data-driven evolutionary algorithm with perturbation-based ensemble surrogates. *IEEE Trans Cybern* 51:3925–3937
- Cui M, Li L, Zhou M, Abusorrah A (2021) Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems. *IEEE Trans Evol Comput* 1:1
- Xiaodong R, Daofu G, Zhigang R, Yongshen L, An C (2021) Enhancing hierarchical surrogate-assisted evolutionary algorithm for high-dimensional expensive optimization via random projection. *Complex Intell Syst* 7(2961–2975):07
- Mohamed AB, Nathalie B, Regis Rommel G, Abdelkader O, Joseph M (2018) Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. *Eng Optim* 50:2038–2053
- Tian J, Tan Y, Zeng J, Sun C, Jin Y (2019) Multiobjective infill criterion driven Gaussian process-assisted particle swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 23:459–472
- Liu B, Akinsolu MO, Song C, Hua Q, Excell P, Xu Q, Huang Y, Imran MA (2021) An efficient method for complex antenna design based on a self adaptive surrogate model-assisted optimization technique. *IEEE Trans Antennas Propag* 69:2302–2315
- Davarynejad M, Ahn CW, Vrancken J, van den Berg J, Coello Coello CA (2010) Evolutionary hidden information detection by granulation-based fitness approximation. *Appl Soft Comput* 10:719–729
- Tian J, Zeng J, Tan Y, Sun C (2018) Adaptive information granulation in fitness estimation for evolutionary optimization. *Wirel Pers Commun* 103:741–759
- Ong YS, Nair PB, Keane AJ (2003) Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA J* 41:687–696
- Wang H, Jin Y, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans Cybern* 47:2664–2677
- Li F, Shen W, Cai X, Gao L, Gary WG (2020) A fast surrogate-assisted particle swarm optimization algorithm for computationally expensive problems. *Appl Soft Comput* 92:106303
- Yang C, Ding J, Jin Y, Chai T (2020) Off-line data-driven multi-objective optimization: knowledge transfer between surrogates and generation of final solutions. *IEEE Trans Evol Comput* 24:409–423
- Sun C, Jin Y, Cheng R, Ding J, Zeng J (2017) Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 21:644–660
- Pan J-S, Liu N, Chu S-C, Lai T (2021) An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems. *Inf Sci* 561:304–325
- Ong Y-S, Zhou Z, Lim D (2006) Curse and blessing of uncertainty in evolutionary algorithm using approximation. In: *2006 IEEE international conference on evolutionary computation*, pp 2928–2935

37. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, vol 4, pp 1942–1948
38. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10:281–295
39. Qu BY, Nagarathnam SP, Swagatam D (2013) A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Trans Evol Comput* 17:387–402
40. Ran C, Yaochu J (2015) A social learning particle swarm optimization algorithm for scalable optimization. *Inf Sci* 291:43–60
41. Cheng YJR (2015) A competitive swarm optimizer for large scale optimization. *IEEE Trans Cybern* 45:191–204
42. Powell MJD (1990) The theory of radial basis function approximation in 1990. University of Cambridge, Department of Applied Mathematics and Theoretical Physics
43. Gutmann H-M (2001) On the semi-norm of radial basis function interpolants. *J Approx Theory* 111:315–328
44. Gutmann H-M (2001) A radial basis function method for global optimization. *J Global Optim* 19:201–227
45. Jin R, Chen W, Simpson TW (2001) Comparative studies of meta-modelling techniques under multiple modelling criteria. *Struct Multidiscip Optim* 23:1–13
46. Allen DM (1974) The relationship between variable selection and data argumentation and a method for prediction. *Technometrics* 16:125–127
47. Suganthan P, Hansen N, Liang J, Deb K, Chen YP, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization
48. Wu G, Mallipeddi R, Suganthan P (Nov 2016) Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization. Technical report, Nanyang Technological University, Singapore
49. Joaquín D, Salvador G, Daniel M, Francisco H (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1:3–18
50. Mueller J (2014) Matsumoto: the matlab surrogate model toolbox for computationally expensive black-box global optimization problems. Preprint [arXiv:1404.4261](https://arxiv.org/abs/1404.4261)
51. Tian J, Sun C, Zeng J, Yu H, Tan Y, Jin Y (2017) Comparisons of different kernels in kriging-assisted evolutionary expensive optimization. In: 2017 IEEE symposium series on computational intelligence (SSCI), pp 1–8
52. Mlakar M, Petelin D, Tušar T, Filipič B (2015) Gp-demo: differential evolution for multiobjective optimization based on Gaussian process models. *Eur J Oper Res* 243:347–361
53. Ulmer H, Streichert F, Zell A (2003) Evolution strategies assisted by Gaussian processes with improved preselection criterion. In: The 2003 congress on evolutionary computation, 2003. CEC'03, vol 1, pp 692–699

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.